INFORMATICS AND COMPUTATIONAL SCIENCES PROGRAM
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY M.Sc. (IT)
A Choice Based Credit System Effective from (2023-24)
Minimum Total and Course Category wise Credit Requirements

S.No.	Broad Category of Courses	Minimum Credit Requirement
1	Discipline Centric Core Course(DCC)	56
2	Discipline Specific Elective Course(DSE)	12
3	Generic Elective Course (GEC)	8
4	Dissertation ,Project, Report of Field Study(DPR)	20
	Total	96

Level	Semester	Course Type	Course Code	Course Title	Delivery Type Per Week			Total Hours	Credits	Internal Marks	External Marks	Max Marks	Remarks
					L	T	P						
8	I	DCC	MIT8000T	Advanced- Data Structure	L	T	-	60	4	20	80	100	
			MIT8001T	Introduction To Python Programming	L	T	-	60	4	20	80	100	
			MIT8002T	Computer Oriented Numerical Methods	L	T	-	60	4	20	80	100	
			MIT8003T	Object Oriented Programming using C++	L	T	-	60	4	20	80	100	

			MIT8000P	Advanced Data Structure lab	-	-	P	120	4	20	80	100	
			MIT8001P	Python Programming Lab	-	-	P	120	4	20	80	100	
	II	DCC	MIT8004T	Advanced Database Management System	L	T	-	60	4	20	80	100	
			MIT8005T	Design & Analysis of Algorithms	L	T	-	60	4	20	80	100	
			MIT8006T	Advanced Operating Systems	L	T	-	60	4	20	80	100	
			MIT8004P	Advanced Database Management System Lab	-	-	P	120	4	20	80	100	
			MIT8005P	Design & Analysis of Algorithms Lab	-	-	P	120	4	20	80	100	
		GEC	MIT810XT	GEC-A	L	T	-	60	4	20	80	100	
Exit with Certificate in Post Graduate Diploma in Computer Application													
9	III	DCC	MIT9006T	Software Engineering	L	T	-	60	4	20	80	100	
			MIT9007T	Compiler Design	L	T	-	60	4	20	80	100	
		DSE	MIT910XT	DSE-A	L	T	-	60	4	20	80	100	
			MIT910XT	DSE-B	L	T	-	60	4	20	80	100	
			MIT910XP	DSE-C	-	-	P	120	4	20	80	100	
		GEC	MIT910XP	GEC-B	-	-	P	120	4	20	80	100	
	IV	DCC	MIT9008T	Report Writing	L	T	-	30	2	20	80	100	
		DSE	MIT91XXP	DSE-D	-	-	P	540	18	20	80	100	

Discipline Specific Elective Course (DSE/GEC)	Paper	Credits	Total Hours
GEC-A			
MIT8100T	Advanced Web Development	4	60
MIT8101T	Artificial Intelligence and Machine Learning	4	60
MIT8102T	Internet of Things	4	60
MIT8103T	Computer Graphics	4	60
GEC-B			
MIT9106P	Cloud Computing Lab	4	120
MIT9107P	Swift Programming Lab	4	120
DSE Group –A			
MIT9104T	Java Programming	4	60
MIT9105T	Data Mining and Data Warehousing	4	60
DSE Group –B			
MIT9106T	Cloud Computing	4	60
MIT9107T	Swift Programming	4	60
DSE Group –C			
MIT8104P	Java Programming Lab	4	120
MIT8105P	Data Mining and Data Warehousing Lab	4	120
DSE Group -D			
MIT9108P	Project Based on Web Technology	20	560
MIT9109P	Project Based on C / C++ / Java / Python	20	560
MIT9110P	Project Based on Mobile Application Programming (Eg. Android)	20	560
MIT9111P	Project Based on Swift Programming	20	560

Semester I

Code of the Course : MIT8000T
Title of the Course : Advanced Data Structure
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science
Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:

1. Induction/recursion, asymptotic notation, recurrence relations.
2. Data structures: linked lists, stacks, queues, binary trees, heaps.
3. Algorithms: binary search, sorting.

Objectives of the Course:

Students will be able to understand the storage, organization, and management of data and information for efficient, easy accessibility and modification of data.

Learning Outcomes:

1. Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.
2. Master a variety of advanced abstract data types (ADT) and data structures and their implementations.
3. Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

Syllabus

UNIT I

Priority Queues (Heaps) – Model, Simple implementations, Binary Heap: Structure-Property, Heap Order Property, And Basic Heap Operations: insert, delete, Percolate down, Other Heap Operations. (12 Lectures Hours)

UNIT II

Trees: Threaded trees and advantages, Applications: Decision trees, Game trees, and expression parsing. Red-Black Trees – Properties of red-black trees, Rotations, Insertion, Deletion.

Multi-way Search Trees – 2-3 Trees: Searching for an Element in a 2-3 Tree, Inserting a New Element in a 2-3 Tree, Deleting an Element from a 2-3 Tree.

(12 Lectures Hours)

UNIT III

Graphs: Graphs and their representations: Matrix representation,List structure,Graph traversal algorithm (DFS and BFS), Application of graphs.Single Source Shortest Path Algorithms: Dijkstra's, Bellman-Ford, All-Pairs Shortest Paths: Floyd-Warshall's Algorithm. (12 Lectures Hours)

UNIT IV

Strings and their features: Strings-Representation and Manipulation using Arrays and lists-string matching algorithms. Brute force, Knuth-Morris-Pratt, and Boyer-Moore strategies. The Huffman Coding Algorithm, Longest Common Subsequence Problem (LCS)
Tables: Decision tables-Symbol tables-Hash Tables-Examples of representation and implementation and Applications. (12 Lectures Hours)

UNIT V

Disjoint Sets – Equivalence relation, Basic Data Structure, Simple Union and Find algorithms, Smart Union and Path compression algorithm.
Review of Hashing: Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing (12 Lectures Hours)

Text and Reference Books:

1. Data Structure & Algorithms, Aho A.V. & Ullman J.E.
2. Introduction to Data Structures, Bhagat Singh & Thomas Naps.
3. Introduction to Algorithms, Charles Leiserson, Ronald Rivest, & Thomas H. Cormen, MIT Press

Suggested E-resources.

(1) https://www.geeksforgeeks.org/priority-queue
(2) https://www.tutorialspoint.com/data_structures_algorithms/red_black_trees.htm
(3) https://www.simplilearn.com/tutorials/data-structure-tutorial/bellman-ford-algorithm

Semester I

Course Code : MIT8001T
Title of the Course : Introduction to Python Programming
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:

1. Experience with a high-level language (C/C++, Java) is suggested. Prior knowledge of scripting
2. language (Perl, UNIX/Linux shells) and Object-Oriented concepts are helpful but not mandatory.

Objectives of the Course: The learning objectives of this course are:

1. To learn how to design and program Python applications.
2. To learn how to use lists, tuples, and dictionaries in Python programs.
3. To learn how to identify Python object types.
4. To define the structure and components of a Python program.
5. To learn how to write loops and decision statements in Python.
6. To learn how to write functions and pass arguments in Python.
7. To learn how to build and package Python modules for reusability.
8. To learn how to read and write files in Python.
9. To learn how to design object-oriented programs with Python classes.
10. To learn how to use exception handling in Python applications for error handling.

Learning Outcomes:

1. Interpret the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements.
2. Express proficiency in the handling of strings and functions.
3. Determine the methods to create and manipulate Python programs by utilizing data structures like lists, dictionaries, tuples, and sets.
4. Identify the commonly used operations involving file systems and regular expressions.
5. Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance, and polymorphism as used in Python.

Syllabus

UNIT I

INTRODUCTION TO PYTHON

Python installation, Python syntax, Scripts, Native Data Types, Booleans, Numbers, Lists, Tuple, Sets, Dictionaries, Comprehensions, List Comprehensions, Dictionary Comprehensions, Set Comprehensions (12 Lectures Hours)

UNIT II

STRINGS AND MODULES

String operation, Formatting, Bytes, Encoding, Regular Expressions, Verbose, module declaration, Importing modules, Objects, and Indenting as Requirement, Exceptions, Unbound Variables, Lambda Functions and map (12 Lectures Hours)

UNIT III

CLASSES

Creating classes, instance methods, Instance Variables, Closures, Generators, Iterators, Assert, Generator Expressions (12 Lectures Hours)

UNIT IV

TESTING AND FILES

Reading and Writing Text Files, Binary Files, Stream Objects, Standard Input, Output, and Error modes, with statements (12 Lectures Hours)

UNIT V

GUI IN PYTHON

Components and events, root component, entry widgets, test widgets, check buttons, Serializing Objects, Pickle Files, Debugging, Introduction to Django framework

(12 Lectures Hours)

Text and Reference Books:

1. Dive into Python,Mark Pilgrim,Press,
2. Python: The Complete Reference, Martin C. Brown
3. Fluent Python: Clear, Concise, and Effective Programming (1st Edition), Luciana Ramalho, OReilly
4. Learning Python: Powerful Object-Oriented Programming, Mark Lutz, O'Reilly, Shroff Publishers And Distributors.

Suggested E-resources.

(1) https://www.w3schools.com/python/python_intro.asp

(2) https://vegibit.com/python-string-module/#:~:text=A%3A%20The%20string%20module%20is,for%20string%20substitution%20using%20templates.

(3) https://realpython.com/python-gui-tkinter/

Semester I

Course Code : MIT8002T
Title of the Course : Computer-Oriented Numerical Method
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science
Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:

(1) High level math:
(2) Calculus.
(3) differential equations.
(4) infinite series.
(5) statistics and probability.
(6) Basic programming language such as C++, Java...

Objectives of the Course:

(1) Obtain an intuitive and working understanding of numerical methods for the basic problems of numerical analysis.
(2) Gain experience in the implementation of numerical methods using a computer.
(3) Trace errors in these methods and need to analyze and predict it.

Learning Outcomes:

(1) Demonstrate understanding of common numerical methods and how they are used to obtain
(2) approximate solutions to otherwise intractable mathematical problems.
(3) Apply numerical methods to obtain approximate solutions to mathematical problems.

Syllabus

UNIT I

Solution of algebraic and transcendental equations — Fixed point iteration method — Newton Raphson method — Solution of linear system of equations — Gauss elimination method — Pivoting — Gauss Jordan method — Iterative methods of Gauss Jacobi and Gauss-Seidel — Eigenvalues of a matrix by Power method and Jacobi's method for symmetric matrices.

(12 Lectures Hours)

UNIT II

Interpolation with unequal intervals — Lagrange interpolation — Newton's divided difference interpolation — Cubic Splines — Difference operators and relations — Interpolation with equal intervals — Newton's forward and backward difference formulae.

(12 Lectures Hours)

UNIT III

Approximation of derivatives using interpolation polynomials — Numerical integration using Trapezoidal, Simpson's 1/3 rule — Romberg's Method — two point and three point Gaussian quadrature formulae — Evaluation of double integrals by Trapezoidal and Simpson's 1/3 rules.
(12 Lectures Hours)

UNIT IV

Single step methods — Taylor's series method — Euler's method — Modified Euler's method — Fourth order Runge — Kutta method for solving first order equations — Multi step methods — Milne's and Adams — Bash forth predictor-corrector methods for solving first-order equations. (12 Lectures Hours)

UNIT-V

Numerical Differentiation, Numerical Integration, Euler-Maclaurin Formula, Adaptive Quadrature Methods, Gaussian Integration, Singular Integrals, Fourier Integrals, Numerical Double Integration (12 Lectures Hours)

Text and Reference Books:

(1) Niyogi, Pradip, "Numerical Analysis and Algorithms", Tata McGraw –Hill
(2) Balagurusamy,E., "Numerical Methods", Tata McGraw –Hill
(3) Sastry, S.S., "Introduction Methods of Numerical Analysis", PHI

Suggested E-resources.

(1) https://books.google.co.in/books?id=FC57DwAAQBAJ&printsec=copyright&redir_esc=y#v=onepage&q&f=false
(2) https://www.ddegjust.ac.in/2021/bca/Computer%20Oriented%20Numerical%20Methods.pdf
(3) https://www.studocu.com/in/document/chandigarh-university/software-engineering/computer-oriented-numerical-methods/39967211

Semester I

Course Code : MIT8003T

Title of the Course : Object-Oriented Programming C++

Qualification Level of the Course : NHEQF Level 6

Credit of the Course : 4

Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite :

Study C Programming concepts for this programming subject.

Objectivesof theCourse:

Object-oriented programming (OOP) is a way of thinking about and organizing code for maximum reusability. With this type of programming, a program comprises objects that can interact with the user, other objects, or other programs. This makes programs more efficient and easier to understand.

LearningOutcomes:

1. Understand the principles and concepts of Object-Oriented Programming (OOP) and its advantages over procedural programming. Differentiate between OOP and procedural programming paradigms. Apply the principles of abstraction, encapsulation, inheritance, and polymorphism in problem-solving.

2. Demonstrate proficiency in C++ programming language basics, including the structure of a C++ program, data types, variables, expressions, operators, pointers, arrays, strings, flow control statements, functions, and memory management. Implement recursive functions, inline functions, and dynamic memory allocation.

3. Design and implement classes and data abstraction in C++. Understand class definition, objects, scope, constructors, destructors, static members, constant member functions, friend functions, and dynamic creation and destruction of objects. Apply function overloading and operator overloading techniques.

Syllabus

UNIT – I

Different paradigms for problem-solving, need for OOP, differences between OOP and procedure-oriented programming, abstraction, overview of OOP principles-encapsulation, inheritance, and data binding polymorphism, abstraction.

C++ basics: structure of a C++ program, data types, declaration of variables, expressions, operators, type conversions, pointers and arrays, strings, structures,

references, flow control statement, functions-scope of variables, parameter passing, recursive functions, default arguments, inline functions, dynamic memory allocation, and deallocation operators. (12 Lectures Hours)

UNIT – II

C++ classes and data abstraction: class definition, class structure, class objects, class scope, this pointer, static class members, constant member functions, constructors and destructors, dynamic creation and destruction of objects, friend function and class, static class member.

Overloading: function overloading, operator overloading – unary, binary operators.

(12 Lectures Hours)

UNIT – III

Inheritance: defining a class hierarchy, different forms of inheritance, defining the base and derived classes, access to the base class members, base and derived class construction, destructors, and virtual base class.

Polymorphism: static and dynamic bindings, base and derived class virtual functions, dynamic binding through virtual functions, virtual function call mechanism, pure virtual functions, abstract classes, implications of polymorphic use of classes, virtual destructors. (12 Lectures Hours)

UNIT – IV

Templates - function templates and class templates, overloading of a function template, a static class member in class template.

Exception handling: benefits of exception handling, throwing an exception, the try block, catching an exception, exception objects, exception specifications, rethrowing an exception, catching all exceptions. (12 Lectures Hours)

UNIT-V

File handling: stream classes hierarchy, stream I/O, file streams, opening and closing data file, creating a data file, read and write functions, error handling during file operations, formatted I/O, sequential and random file processing.

Standard template library (STL): component of STL, containers, iterators, algorithms, and application of container classes. (12 Lectures Hours)

Text and References Books:

1. Object-Oriented Programming with C++: E. Balagurusamy
2. C++: The Complete Reference, Herbert Schildt,McGraw Hill.
3. Let Us C++, Yashwant Kanetkar, Bpb Publisher.
4. C C++ C Programming Concepts Black Book, Dasgupta, Chakrabarti, Dreamtech Press

Suggested E-resources..

1. NPTEL
2. Coursera

Semester I

Course Code : MIT8000P
Title of the Course : Advanced Data Structure Lab

Qualification Level of the Course : NHEQF Level 6

Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science
Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:

(1) Induction/recursion, asymptotic notation, recurrence relations.
(2) Data structures: linked lists, stacks, queues, binary trees, heaps.
(3) Algorithms: binary search, sorting.

Objectives of the Course:

Students will be able to understand the storage, organization, and management of data and information for efficient, easy accessibility and modification of data.
Learning Outcomes:

(1) Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.
(2) Master a variety of advanced abstract data types (ADT) and data structures and their implementations.
(3) Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

List of few programs:

(1) To implement functions of Dictionary using Hashing (division method, Multiplication method,
(2) Universal hashing)
(3) To perform various operations i.e., insertions and deletions on 2-3 trees.
(4) To implement operations on binary heap.
(5) To implement operations on graphs
(6) To implement Krushkal's algorithm to generate a min-cost spanning tree.
(7) To implement Dijkstra's algorithm to find the shortest path in the graph.
(8) To implement pattern matching using the Boyer- Moore algorithm.
(9) To implement the Knuth-Morris-Pratt algorithm for pattern matching.
(10) To perform various operations i.e., insertions and deletions on AVL trees
For given expression eg. a-b*c-d/e+f
(11) construct in order sequence and traverse it using post order traversal(non recursive)
(12)Programs based on Tree
(13)Programs based on Graph

Text and Reference Books:

(1) Data Structure & Algorithms, Aho A.V. & Ullman J.E.
(2) Introduction to Data Structures, Bhagat Singh & Thomas Naps.
(3) Introduction to Algorithms, Charles Leiserson, Ronald Rivest, & Thomas H. Cormen, MIT Press

Suggested E-resources.

(1) https://www.geeksforgeeks.org/priority-queue
(2) https://www.tutorialspoint.com/data_structures_algorithms/red_black_trees.htm
(3) https://www.simplilearn.com/tutorials/data-structure-tutorial/bellman-ford-algorithm

Scheme of EoSE: - 1) Exercise(s): 45

2) Viva Voce: 20

3) Evaluation of record book: 15

Semester I

Code of the Course **:**MIT8001P
Title of the Course : Python Programming Lab
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course : Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:

(1) Experience with a high level language (C/C++, Java) is suggested. Prior knowledge of a scripting
(2) language (Perl, UNIX/Linux shells) and Object-Oriented concepts are helpful but not mandatory.

Objectives of the Course: The learning objectives of this course are:

(1) To learn how to design and program Python applications.
(2) To learn how to use lists, tuples, and dictionaries in Python programs.
(3) To learn how to identify Python object types.
(4) To define the structure and components of a Python program.
(5) To learn how to write loops and decision statements in Python.
(6) To learn how to write functions and pass arguments in Python.
(7) To learn how to build and package Python modules for reusability.
(8) To learn how to read and write files in Python.
(9) To learn how to design object-oriented programs with Python classes.
(10) To learn how to use exception handling in Python applications for error handling.

Learning Outcomes:

(1) Interpret the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements.
(2) Express proficiency in the handling of strings and functions.
(3) Determine the methods to create and manipulate Python programs by utilizing data structures like lists, dictionaries, tuples, and sets.
(4) Identify the commonly used operations involving file systems and regular expressions.
(5) Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance, and polymorphism as used in Python.

List of few programs:

(1) Write a Python program/ script to create a list and perform the following methods 1) insert() 2) remove() 3) append() 4) len() 5) pop() 6) clear()
(2) Write a Python program/ script to create a dictionary and apply the following methods 1) Print the dictionary items 2) access items 3) use get() 4)change values 5) use len()

(3) Write a Python program/ script to create a tuple and perform the following methods 1) Add items 2) len() 3) Check for item in tuple 4)Access items
(4) Write a Python program to add two numbers, to print a number is positive/negative using if-else, to find the largest number among three numbers.
(5) Write a Python program to read a number and display the corresponding day using if_elif_else.
(6) Write a menu driven program to create with the following options 1. TO PERFORM ADDITION 2. TO PERFORM SUBTRACTION 3. TO PERFORM MULTIPLICATION 4. TO PERFORM DIVISION
(7) Python Program to Implement Conditionals and Loops
(8) Python program to use functions and modules
(9) Python program to read and write data from & to files in Python
(10) Python program based on Classes and Objects
(11) Python program based on inheritance
(12) Python program based on lambda Function

Text and Reference Books:

(1) Dive into Python,Mark Pilgrim,Press,
(2) Python: The Complete Reference, Martin C. Brown
(3) Fluent Python: Clear, Concise, and Effective Programming (1st Edition), Luciana Ramalho, OReilly
(4) Learning Python: Powerful Object-Oriented Programming, Mark Lutz, O'Reilly, Shroff Publishers And Distributors.

Suggested E-resources.

(1) https://www.w3schools.com/python/python_intro.asp
(2) https://vegibit.com/python-string-module/#:~:text=A%3A%20The%20string%20module%20is,for%20string%20substitution%20using%20templates.
(3) https://realpython.com/python-gui-tkinter/

Scheme of EoSE: - 1) Exercise(s): 45 2) Viva Voce: 20 3) Evaluation of record book: 15

Semester II

Course Code : MIT8004T
Title of the Course : Advanced Database Management System
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.
Prerequisites **:** NONE

Objectives of the Course:

1. This Course helps students in understanding the concepts of databases
2. This Course helps students in Modelling the databases with different types of models available.
3. This Course helps students in designing the databases.
4. This course helps students to learn about how to store and retrieve the data from databases

Course Learning Outcomes:

(1) Learn about different features of database management systems.
(2) Differentiate between database systems and file systems.
(3) Model a database system using modeling tools like ER diagrams
(4) Design database schemas based on the conceptual model.
(5) Write queries in relational algebra / SQL.
(6) Normalize a given database schema

Syllabus

UNIT I

Introduction: Database system applications, database systems versus file systems, views of data, data models, database languages, database users and administrators, transaction management, database system structure, and application architecture.Data modeling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, concepts of super key, candidate key, primary key, unique key, generalization, aggregation, reduction of an ER diagram to tables. (12 Lectures Hours)

UNIT II

Relational model: Structure of relational databases, relational algebra, tuple relational calculus, domain relational calculus.SQL: Characteristics of SQL, advantages of SQL, types of SQL commands, SQL operators and their procedure, tables, views and indexes, queries and sub-queries, aggregate functions, insert, update and delete operations, joins, union, intersection, minus, cursors in SQL. Domain constraints, referential integrity, assertions, triggers, authorization, and authentication.Relational database design & normalization: Functional

dependencies, normal forms- First, second,third, BCNF, fourth and fifth normal forms, decomposition. (12 Lectures Hours)

UNIT III

Indexing and Hashing: Basic concepts, ordered indices, B-tree, B+ tree, static hashing, dynamic hashing, comparison of ordered indexing and hashing, index definition in SQL, multiple-key access. Query Processing & Optimization: Measure of query cost, selection operation, sorting, join operation, and other operations (12 Lectures Hours)

UNIT IV

Transactions: Transaction concept, atomicity, durability, concurrent execution, serializability, conflict and view, testing of serializability.

Concurrency Control: Concurrency Control, Locking Techniques for Concurrency control, Time stamping protocols for concurrency control, validation-based protocols,

Recovery System: Failure classification, storage structure (RAID), recovery and atomicity, log-based recovery, shadow paging (12 Lectures Hours)

UNIT V

Object Oriented Database Concept: Data types and Object, Evolution of Object-Oriented Concepts, Characteristics of Object-Oriented Data Model. Object Hierarchies, Generalization, Specialization, Aggregation. Object Schema. Inter-object Relationships, Similarities and difference between Object Oriented Database model and Other Data models.

Object Oriented DBMS Architecture, Application Selection for Object Oriented DBMS, Database Design for an Object Relational DBMS. Data Access API (ODBC, DAO, ADO, JDBC, OLEDB) (12 Lectures Hours)

Text and Reference Books:

(1) Database Systems Concepts, Korth
(2) Fundamental of database system - Elmasiri and Navathe
(3) Database Systems, Date C.J., AddisionWesley
(4) DBM and Design, Hansen and Hansen, PHI
(5) Distributed Databases, Ceri S, Pelagatti G, Principles and Systems,McGraw Hill.

Suggested E-Resources:

(1) Date, C.J., Kanman, A. & Swamynathan, S. (2006). An Introduction to Database Systems. 8th edition. Pearson Education.
(2) Ramakrishnan, R. Gehrke, J. (2014), Database Management Systems. 3rd edition. Tata McGraw Hill Education.
(3) MWidenius, M., Axmark, D., Cole, J., Lentz, A., & Dubois, P. (2002). MySQL Reference Manual O reilly community press

Semester II

Course Code : MIT8005T
Title of the Course : Design & Analysis of Algorithms
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:

The Students should have basic knowledge of programming and mathematics. The Students should know data structure very well.

Objectives of the Course:

(1) Analyze the asymptotic performance of algorithms.
(2) Write rigorous correctness proofs for algorithms.
(3) Demonstrate a familiarity with major algorithms and data structures.
(4) Apply important algorithmic design paradigms and methods of analysis.

Learning Outcomes:

(1) Argue the correctness of algorithms using inductive proofs and invariants.
(2) Analyze worst-case running times of algorithms using asymptotic analysis.
(3) Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it.
(4) Describe the dynamic programming paradigm and explain when an algorithmic design situation calls for it.
(5) Describe the greedy paradigm and explain when an algorithmic design situation calls for it.
(6) Explain the major graph algorithms and their analyses
(7) Explain the matrices and their applications
(8) Understand the use of finite automata

Syllabus

UNIT-I

Algorithms Analysis: Algorithms and structured programming. Analyzing algorithms, asymptotic behavior of an algorithm, recurrence relation, Order notations, time and space complexities, average and worst case analysis, lower and upper bounds.

(12 Lectures Hours)

UNIT-II

Algorithm design strategies: Divide and conquer (Merge sort, Quick sort, matrix multiplication), Greedy method (knapsack problem, minimum spanning trees).

Basic search & Traversal Techniques (Breadth first and Depth first traversals of Graphs). (12 Lectures Hours)

UNIT-III

Dynamic programming: 0/1 knapsack, Travelling salesman problem Backtracking: 8-queen problem, sum of subsets, 0/1 Knapsack

Branch & Bound: 0/1 knapsack, Travelling salesman.

(12 Lectures Hours)

UNIT-IV

Matrix algorithms: Basics of matrices, Strassen's matrix-multiplication algorithm

Data structures for set manipulation problems: Fundamental operation on sets, a simple disjoint- set union algorithm, tree structures for UNION-FIND problem, applications and extensions of the UNION-FIND algorithm.

(12 Lectures Hours)

UNIT-V

Finite automata and regular expression, recognition of regular expression, patterns, recognition of substrings, Conversion from NFA to DFA

Complexity Theory: Overview, Turing machine, polynomial and non-polynomial problems, deterministic and non-deterministic algorithms, Idea of problem Classes: P class, NP class & NP complete problems

(12 Lectures Hours)

Text and Reference Books:

(1) Fundamentals of Computer Algorithms, E. Horowitz, S. Sahni, Galgotia Publications.

(2) Introduction to Algorithms, Charles Leiserson, Ronald Rivest, & Thomas H. Cormen, MIT Press

(3) Design & Analysis of Computer Algorithms, Av. Aho, J.E. Hopcroft, & J.D. Ullman, Addition Wesley.

(4) Design and Analysis of algorithms, S.K. Basu, PHI Publications

Suggested E-resources

(1) https://www.javatpoint.com/daa-tutorial

(2) https://www.javatpoint.com/dynamic-programming

Semester II

Course Code : MIT8006T
Title of the Course : Advanced Operating Systems
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course : Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.
Prerequisite : None

Course Objectives:

(1) To understand the need of Operating Systems
(2) To understand the Roles of an Operating System
(3) To understand how these Roles are performed by an Operating System

Learning Outcomes:

(1) Understanding systems concepts like virtualization
(2) Comprehend how an operating system virtualises CPU and memory.

Syllabus

UNIT I

Introduction to Operating Systems: Mainframe systems, desktop systems, multiprocessor systems, distributed systems, clustered systems, real-time systems, handheld systems. Feature migration and computing Environments. Computer System Structures: Computer system operation, I/O structure, storage structure, storage hierarchy, hardware protection, network structure. Operating System Structures: System components, operating system services. System calls, system programs, system structure, virtual machines.

(12 Lectures Hours)

UNIT II

Processes: Process concept, process scheduling, operations on processes, cooperating processes, Inter-process communication, communication in client-server systems. Threads: Overview, multithreading models, threading issues.

(12 Lectures Hours)

UNIT III

CPU Scheduling: Basic Concepts, scheduling criteria, scheduling algorithms, multiple-processor scheduling, real-time scheduling, algorithm evaluation. Process Synchronization: The critical section problem, synchronization hardware, semaphores, classical problems of synchronization, monitors. Deadlocks: System model, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

(12 Lectures Hours)

UNIT IV

Storage and Memory Management: Swapping, contiguous memory allocation, paging, segmentation, segmentation with paging.

Virtual Memory: Demand paging, process creation, page replacement, allocation of frames, thrashing. File System Interface: File concept, access methods, directory structure, file system mounting, file sharing, protection.

File-System Implementation: File system structure, file-system implementation, directory implementation, allocation methods, free space management, efficiency and performance. (12 Lectures Hours)

UNIT V

Protection: Goals of protection, domain of protection, access matrix, implementation of access matrix, revocation of access rights.

Security: The security problem, user authentication, program threats, system threats, security systems and facilities, intrusion detection, cryptography.

(12 Lectures Hours)

Text and Reference Books:

(1) Operating System Concepts, Silberschatz G.G., John Wiley &SonsInc.
(2) Modern Operating Systems, Andrew S. Tanenbaum,Pearson Prentice Hall,
(3) Advanced Concepts in Operating Systems Distributed, Database, and Multiprocessor Operating Systems, Mukesh Singhal and Niranjan G. Shivaratri, Tata McGraw-Hill
(4) Operating Systems: A Concept-based Approach, Dhananjay M. Dhamdhere, Tata McGraw-Hill Education.
(5) Distributed Systems: Concepts and Design, Coulouris et al, Addison Wesley.
(6) Tanenbaum and Steen: Distributed Systems: Principles and Paradigms, Pearson Education

Suggested E-resources.

(1) https://www.javatpoint.com/os-cpu-scheduling
(2) https://www.tutorialspoint.com/operating_system/os_memory_management.htm
(3) https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/14_Protection.html

Semester II

Code of the course: MIT8004P
Title of the course: Advanced Database Management System Lab
Level of the Course: NHEQF Level 6
Credit of the Course: 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites: NONE

Objectives of the Course: The learning objectives of this course are:

(1) This Course helps students in implementing the concepts of databases learned in theory.
(2) This Course helps students in modelling the databases with different types of models available.
(3) This course helps students to perform the operations to store and retrieve the data from databases

Learning Outcomes:On successful completion of the course, students will be able to:

(1) Learn about different features of database management systems.
(2) Differentiate between database systems and file systems.
(3) Model a database system using modeling tools like ER diagrams
(4) Design database schemas based on the conceptual model.
(5) Write queries in relational algebra / SQL.
(6) Normalize a given database schema.

List of few programs:

1. Create a database having two tables with the specified fields, to computerize a library system of a MLSU.
LibraryBooks (Accession number, Title, Author, Department, PurchaseDate, Price)
IssuedBooks (Accession number, Borrower)
a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
b) Delete the record of a book titled "Database System Concepts".
c) Change the Department of the book titled "Discrete Maths" to "CS"
d) List all books that belong to the "CS" department.
e) List all books that belong to the "CS" department and are written the by author "Navathe".
f) List all computer (Department = "CS") that have been issued.
g) List all books which have a price less than 500 or purchased between "01/01/1999" and "01/01/2004".

2. Create a database table having three tables to store the details students of the computer Department in your college, as per the given schema.
Personal information about the Student (College roll number, Name of student, Date of birth, Address, Marks(rounded off to whole number) in percentage at 10 + 2, Phone

number)Paper Details (Paper code, Name of the Paper)Student Academic and Attendance details (College roll number, Paper code, Attendance, Marks in-home examination).

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.

b) Design a query that will return the records (from the second table) along with the name of the student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper 2.

c) List all students who live in "Delhi" and have marks greater than 60 in paper 1.

d) Find the total attendance and total marks obtained by each student.

3. Create the following tables, enter at least 5 records in each table and answer the queries given below.

EMPLOYEE (Person_Name, Street, City)
WORKS (Person_Name, Company_Name, Salary)
COMPANY (Company_Name, City)
MANAGES (Person_Name, Manager_Name)

a) Identify primary and foreign keys.

b) Alter table emp

c) Find the name of all managers who work for both Samba Bank and NCB Bank.

d) Find the names, street address and cities of residence and salary of all employees who earn more than $10,000.

e) Find the names of all employees who live in the same city as the company for which they work.

f) Find the highest salary, lowest salary and average salary paid by each company.

g) Find the sum of salary and number of employees in each company.

h) Find the name of the company that pays highest salary.

e) List the name of student who has got the highest marks in paper

2. Create the following tables, enter at least 5 records in each table and answer the queries given below.

Suppliers (SNo, Sname, Status, SCity)
Parts (PNo, Pname, Colour, Weight, City)
Project (JNo, Jname, Jcity)
Shipment (Sno, Pno, Jno, Quantity)

a) Identify primary and foreign keys.

b) Get supplier numbers for suppliers in Paris with status>20.

c) Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.

d) Get suppliers names for suppliers who do not supply part P2.

e) For each shipment get full shipment details, including total shipment weights.

f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.

g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.

h) Get the names of cities that store more than five red parts.

i) Get full details of parts supplied by a supplier in London.

j) Get part numbers for parts supplied by a supplier in London to a project in London.
k) Get the total number of projects supplied by a supplier (say, S1).
l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).

Text and Reference Books:

(1) Database Systems Concepts, Korth
(2) Fundamentals of database system - Elmasiri and Navathe
(3) Database Systems, Date C.J., Addison-Wesley
(4) DBM and Design, Hansen and Hansen, PHI
(5) Distributed Databases, Ceri S, Pelagatti G, Principles and Systems,McGraw Hill.

Suggested E-Resources:

(1) Date, C.J., Kanman, A. & Swamynathan, S. (2006). An Introduction to Database Systems. 8th edition. Pearson Education.
(2) Ramakrishnan, R. Gehrke, J. (2014), Database Management Systems. 3rd edition. Tata McGraw Hill Education.
(3) MWidenius, M., Axmark, D., Cole, J., Lentz, A., & Dubois, P. (2002). MySQL Reference Manual O Reilly community press

Scheme of EoSE: -

1) Exercise(s): 45

2) Viva Voce: 20

3) Evaluation of record book: 15

Semester II

Course Code : MIT8005P
Title of the Course : Design & Analysis of Algorithms Lab
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:
Proficiency in a C & C++ programming language, basic program design concepts (e.g, pseudocode), proof techniques, familiarity with trees and graph data structures, familiarity with basic algorithms such as those for searching, and sorting
Objectives of the Course: The learning objectives of this course are:

(1) To identify and analyze worst-case running times of algorithms.
(2) To model a given engineering problem using graphs and trees and write the corresponding algorithm to solve the problems.
(3) To strengthen the ability to identify and apply a suitable algorithm for the given real-world problem.

Learning Outcomes:After Completing this course, the students will be able to:

(1) Design an algorithm in an effective manner
(2) Apply iterative and recursive algorithms.
(3) Design iterative and recursive algorithms
(4) Implement optimization algorithms for specific applications.
(5) Design optimization algorithms for specific applications

List of Some Programs for reference:

1. Implement Merge Sort
2. Implement Quick Sort
3. Implement Binary Search
4. Implement DFS and BFS traversals
5. Implement knapsack problem
6. Implement N Queens Problem
7. Implement minimum cost spanning tree
8. Implement all pair shortest path
9. Implement Strassen Multiplication

Text and Reference Books:

1. Fundamentals of Computer Algorithms, E. Horowitz, S. Sahni, Galgotia Publications.
2. Introduction to Algorithms, Charles Leiserson, Ronald Rivest, & Thomas H. Cormen, MIT Press
3. Design & Analysis of Computer Algorithms, Av. Aho, J.E. Hopcroft, & J.D. Ullman, Addition Wesley.
1. Design and Analysis of algorithms, S.K. Basu, PHI Publications

Suggested E-resources

1. https://www.javatpoint.com/daa-tutorial
2. https://www.javatpoint.com/dynamic-programming
3. https://www.codingninjas.com/studio/library/regular-expression-and-finite-automata#:~:text=Finite%20automata%20are%20used%20to,language%20accepted%20by%20Finite%20Automata.

Scheme of EoSE: - 1) Exercise(s): 45
2) Viva Voce: 20
3) Evaluation of record book: 15

GEC Group – A

Semester II

Course Code : MIT8100T
Title of the Course **:**Advanced Web Development
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course : Generic Elective Course (GEC) in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:
To become a Web Developer, you should have an understanding of HTML, CSS, and JavaScript. It's also recommended to learn about CSS and CSS frameworks.

Objectives of the Course : The learning objectives of this course are:

(1) The increasing practice of Web-based applications, this course focuses on Advanced Web Technologies with the concept of Laravel Framework
(2) This subject will attempt to give basic understanding of HTML, CSS, Javascript, URL Methods, MVC Framework, PHP, Laravel

Learning Outcomes:

(1) Develop a fully functioning website and deploy on a web server.
(2) Find and use code packages based on their documentation to produce working results in a project.
(3) Create webpages that function using external data.
(4) Architect solutions to programming problems by combining visual components and classes.

Syllabus

UNIT-I

Introduction: Basics of WWW, HTTP protocol methods and headers, HTTP Request and Response, Architecture of web browser, Web server installation and configuration, Web security, CORS, Understanding SEO

HTML: HTML page structure, formatting tags in HTML, tables, links, images, meta tags, frames, html form tags, media, APIs, HTML5 tags in relation to validations and SEO

(12 Lectures Hours)

UNIT-II

CSS:NeedforCSS,Basicsyntaxandstructure,Backgrounds,Colorsandproperties,Manipulatingtexts, Fonts,bordersandboxes,Margins,PaddingLists,CSS2,CSS3,Animations, Tool-Tips, Style images, Variables, Media Queries, Wildcard Selectors (*, ^and$)in CSS, MediaQuery, CSSvariables

(12 Lectures Hours)

UNIT-III

JAVA SCRIPT:- Java script Syntax, Types of Java script, variables, arrays, functions, conditions, loops, Popup boxes, Java script objects and DOM, Java script inbuilt functions, Java script validations, Regular expressions, Event handling with Java script, Callbacks in Java script, Function as arguments in Java script, Object concepts in Java script, JSON

(12 Lectures Hours)

UNIT-IV

Php Programming : Introduction to Server side programming , PHP variables, decision and looping with examples, PHP and HTML, Arrays, Functions, Browser control and detection, String, Form processing, File uploads, Dates and time zone, Working with Regular Expressions, Exception Handling, Working with JSON data, Object Oriented Programming with PHP

(12 Lectures Hours)

UNIT-V

PHP MVC Framework - Laravel Introduction to Laravel and MVC, Environment Setup, Routes, Namespaces, Controllers, Blade template, Views, Request Response, Redirections, Forms, Session, Cookies, Migration, Error handling, Database connectivity and CRUD Operations.

(12 Lectures Hours)

Text and Reference Books:

(1) A beginner's guide to HTML, CSS, Javascript, and Web Graphics, by Jennifer Niederst Robbins
(2) MarijnHaverbake's Eloquent Javascript
(3) Kyle Simpson's You Don't Know JS

Suggested E-resources.

(1) https://www.geeksforgeeks.org/mvc-framework-introduction/
(2) https://www.netsolutions.com/insights/what-is-php/
(3) https://learn.microsoft.com/en-us/archive/msdn-magazine/2007/may/advanced-web-applications-with-object-oriented-javascript

GEC Group – A

Semester II

Course Code: MIT8101T
Title of the Course: Artificial Intelligence and Machine Learning
Qualification Level of the Course: NHEQF Level 6
Credit of the Course: 4
Type of the Course: Generic Elective Course (GEC) in Computer Science
Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.
Prerequisite:

(1) Knowledge of Mathematics
(2) Good programming knowledge
(3) Analytical skills
(4) Ability to understand complex algorithms
(5) Basic knowledge of Statistics and modelling
(6) Basics of Algorithms
(7) Basics of Python Programming Language

Objectives of the Course : The learning objectives of this course are:

(1) The learning objective of this course are:
(2) To educate the students of MCA with industrial standards.
(3) To understand the concepts of Artificial Intelligence and Machine Learning (AI&ML)
(4) To elevate innovative research and development in AI & ML to serve as per society needs.
(5) To understand various strategies of generating models from data and evaluating them
(6) To apply ML algorithms on given data and interpret the results obtained
(7) To design appropriate ML solution to solve real world problems in AI domain

Learning Outcomes:

(1) Understand the meaning, purpose, scope, stages, applications, and effects of AI.
(2) Develop a good understanding of fundamental principles of machine learning.
(3) Gain an in-depth understanding of AI problems and techniques.
(4) Gain an in-depth understanding of machine learning algorithms.

Syllabus

UNIT-I

Overview of AI, Problems, Problem space and searching techniques, Definition- production system, Control strategies- forward and backward chaining, Heuristic search techniques-Hill Climbing, Best first Search, A* algorithm, AND/OR Graphs

(12 Lectures Hours)

UNIT-II

Knowledge representation-Propositional Logic, First Order predicate logic, Skolemization, Resolution Principles and unification, Horn clause
Expert System: Introduction, Component, development process. Learning, Planning and Explanation in Expert Systems, Study of existing expert systems: MYCIN & AM

(12 Lectures Hours)

UNIT-III

Machine Learning: Learning, Types of Machine Learning, Learning: Supervised, Unsupervised, Semi supervised and Reinforcement learning. Data source, Curse of dimensionality, Overview of Dimension Reduction

(12 Lectures Hours)

UNIT-IV

Supervised learning: Regression & Classification : Linear regression, Logistic regression, Support Vector Machines, Baye's classifiers, Decision Trees, Ensemble Learning, Nearest Neighbor Methods, Applications

(12 Lectures Hours)

UNIT-V

Neural Architecture: Human brain, Neuron model, Single layer, perceptron, Multiple layer perceptron, Forwards and Backwards Propagation, Error, learning rule. Back propagation: generalized delta rule, Neural network applications.

Unsupervised learning: Clustering, K-means algorithm, Association rules, Applications.

(12 Lectures Hours)

Text and Reference Books:

(1) Artificial Intelligence: Elaine Rich, Kevin Knight, Mc-Graw Hill.
(2) Artificial Intelligence: A Modern Approach. Stuart Russell and Peter Norvig. Prentice Hall.
(3) Introduction to AI & Expert System: Dan W. Patterson, PHI.
(4) Machine Learning, C. Bishop T. M. Mitchell, McGraw-Hill
(5) Pattern Recognition and Machine Learning. Berlin: Springer-Verlag.
(6) Understanding Machine Learning. Shai Shalev-Shwartz and Shai Ben-David. Cambridge University Press. 2017
(7) Foundations of Data Science. Avrim Blum, John Hopcroft and Ravindran Kannan. January 2017.
(8) The Elements of Statistical Learning, by Trevor Hastie, Robert Tibshirani, Jerome H. Friedman

Suggested E-Resources:

(1) https://archive.nptel.ac.in/courses/106/106/106106139/
(2) https://onlinecourses.nptel.ac.in/noc22_cs56/preview
(3) https://onlinecourses.nptel.ac.in/noc23_cs18/preview
(4) https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained
(5) https://www.andrewng.org/courses/

GEC Group – A

Semester II

Course Code : MIT8102T
Title of the Course **:** Internet of Things
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Generic Elective Course (GEC) in Computer Science
Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites:

(1) Must know how systems work.
(2) To know about statistics that has to do with Machine Learning.
(3) Simultaneous use of artificial intelligence
(4) Need to know something about sensors.
(5) Basic Knowledge of Networking protocols.

Objectives of the Course: The learning objectives of this course are:
To obtain and analyze data from things (devices) that were previously disconnected from most data processing tools

Learning Outcomes:

(1) Explain the definition and usage of the term "Internet of Things" in different contexts.
(2) Understand the key components that make up an IoT system.
(3) Differentiate between the levels of the IoT stack
(4) Familiarity with the key technologies and protocols employed at each layer of the IoT stack.
(5) Able to build IoT applications
(6) Learn to integrate modern technologies such as sensors, communication, and computational processing.

Syllabus

UNIT I

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

(12 Lectures Hours)

UNIT II

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

(12 Lectures Hours)

UNIT III

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

(12 Lectures Hours)

UNIT IV

Building IoT With Raspberry Pi & Arduino Building IOT with RASPBERRY PI- IoT Systems - Logical Design using Python,IoT Physical Devices & Endpoints - IoT Device - Building blocks -Raspberry Pi -Board - Linux on Raspberry Pi - Raspberry Pi Interfaces - Programming Raspberry Pi with Python - Other IoT Platforms - Arduino.

(12 Lectures Hours)

UNIT V

Case Studies and Real-World Applications Real world design constraints - Applications - Asset management, Industrial automation, smart grid, Commercial building automation, Smart cities - participatory sensing - Data Analytics for IoT, Software & Management Tools for IoT Cloud Storage Models & Communication APIs - Cloud for IoT - Amazon Web Services for IoT.

(12 Lectures Hours)

Text and Reference Books:

(1) Internet of Things, A hands-on approach ArshdeepBahga, Vijay Madisetti, Universities Press,
(2) Architecting the Internet of Things Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), Springer.
(3) The Internet of Things in the Cloud: A Middleware Perspective Honbo Zhou, CRC Press.
(4) The Internet of Things, Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi, Wiley.

Suggested E-resources.

(1) https://www.elprocus.com/building-the-internet-of-things-using-raspberry-pi/
(2) https://www.techtarget.com/iotagenda/tip/Top-12-most-commonly-used-IoT-protocols-and-standards
(3) https://docbox.etsi.org/workshop/2011/201110_m2mworkshop/02_m2m_standard/m2mwg2_architecture_pareglio.pdf

GEC Group – A

Semester II

Course Code : MIT8103T
Title of the Course **: Computer Graphics**
Qualification Level of the Course : NHEQF Level 6
Credit of the Course : 4
Type of the Course: Generic Elective Course (GEC) in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites: None

Objectives of the Course:

(1) Computer Graphics is used where a set of images needs to be manipulated or the creation of the image in the form of pixels and is drawn on the computer.
(2) Computer Graphics can be used in digital photography, film, entertainment, electronic gadgets, and all other core technologies which are required.

Learning Outcomes:

(1) Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
(2) Understand various algorithms for scan conversion and filling of basic objects and their comparative analysis

Syllabus

UNIT I

Geometry and Line generation: Lines, Line segments and perpendicular lines, distance between apoint and a line, vectors, pixels, frame buffers, vector generation, Bresenham's algorithm, antialiasing of line, thick line segments, character generation, display the frame buffer.Graphics Primitives: Display devices, primitive operations, Display file interpreter, Normalizeddevice co-ordinates, Display file structure and display file algorithms, Display control, text, Linestyle primitives. (12 Lectures Hours)

UNIT II

Polygons: Polygon representation, Entering polygons, Polygon interfacing algorithms, filling polygons, filling with a pattern, Initialization, Antialiasing.
Segments: Creation of segment, Closing, deletion and renaming segments, visibility, image transformations, saving and showing segments.

(12 Lectures Hours)

UNIT III

2D and 3D Transformations: Matrices, Scaling transformations, Rotation, Homogeneous coordinates and Translations, Co-ordinate transformations, Rotation about an arbitrary point, Inverse transformations, Transformation routines, Transformation and patterns, Initialization, Display procedures. 3D geometry, 3D primitives and transformations.

(12 Lectures Hours)

UNIT IV

Windowing and Clipping: The viewing transformation and its implementation, Clipping, Cohen

Sutherland Outcode algorithm, Clipping of polygons, generalized clipping, Multiple windowing,

Parallel projection, Viewing projections and special projections, Conversion to view plane coordinates, Clipping in three dimensions, Clipping planes.

(12 Lectures Hours)

UNIT V

Hidden surfaces and Lines: Back-face algorithm, Z-buffers, Scan line algorithm, Franklin

algorithm, Illumination, Transparency, Reflection, Shadows, Ray tracing, halftones, Color Models

(12 Lectures Hours)

Text and Reference Books:

(1) Computer Graphics, Hearn and Baker, PHI
(2) Computer Graphics: A programming Approach, Steven Harrington
(3) Computer Graphics, principles and practice, Foley, VanDam, Feiner, Hughes, Addison Wesley.
(4) Mathematical Elements for Computer Graphics, David F. Rogers, Adams, McGraw Hill.
(5) Procedural Elements for Computer Graphics, David F. Rogers, McGraw Hill.

Suggested E-resources.

(1) https://epgp.inflibnet.ac.in/Home/ViewSubject?catid=fBYckQKJvP3a/8Vd3L08tQ==
(2) https://nptel.ac.in/courses/106/106/106106090/

Semester III

Course Code : MIT9006T
Title of the Course : Software Engineering
Qualification Level of the Course : NHEQF Level 6.5
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites: None

Objectives of the Course:
The basic objective of software engineering is to develop methods and procedures for software development that can scale up for large systems and that can be used consistently to produce high-quality software at low cost and with a small cycle of time.

Learning Outcomes: The students would be able to understand:

(1) Understanding the Software Engineering Fundamentals, Software development Process with different types of models, Project management Concepts.
(2) Understand the Software Quality Assurance concepts, Software Configuration Management
(3) Understand the Software Quality Assurance concepts, Software Configuration Management, Analysis Concepts and Principles
(4) Get acquainted with Design Concepts and Principles, Software Testing.
(5) Understand the purpose of Reengineering with some CASE Tools

Syllabus

Unit I

Software Engineering Fundamentals: Definition of Software, Software characteristics, Software Applications.Software Process: Software Process Models - Waterfall model, prototyping model, spiral model, incremental model, concurrent development model.Project management Concepts: The Management Spectrum - The People, The Product, The Process, The Project.

(12 Lectures Hours)

Unit II

Software Process and Project Metrics: Measures, Metrics and Indicators, Software measurement: Size - Oriented Metrics, Function - Oriented Metrics, Extended Function point metrics Software Project Planning: Project Planning Objectives, Software Project Estimation, Decomposition Techniques - Problem Based Estimation, Process Based Estimation, Empirical Estimation Models- The COCOMO Model Risk Analysis and Management: Software risks, Risk identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring and Management.

(12 Lectures Hours)

Unit III

Software Quality Assurance: Basic concepts- Quality, Quality Control, Quality Assurance, Cost of Quality, Software Quality Assurance (SQA), Formal Technical Review
Software Configuration Management: Baselines, Software Configuration Items, The SCM Process, Version Control, Change Control, Configuration Audit, Status Reporting.
Analysis Concepts and Principles: Requirements Elicitation for Software, Analysis Principles -The Information Domain, Modeling, Partitioning, Essential and Implementation Views, Specification: Specification Principles, Representation, The Software Requirement Specification (SRS)

(12 Lectures Hours)

Unit IV

Design Concepts and Principles: Design Principles, Design Concepts, Abstraction, Refinement, Modularity, Software Architecture, Control Hierarchy, Structural Partitioning, Data Structure, Software Procedure, Information Hiding, Effective Modular Design-Cohesion, Coupling

(12 Lectures Hours)

Unit V

Software Testing: Testing Objectives & principles, Unit Testing, Integration Testing (Top Down Integration, Bottom Up Integration, Regression Testing, Smoke Testing), Validation Testing (Alpha and Beta Testing), System Testing (Recovery Testing, Security Testing, Stress Testing, Performance Testing).

Reengineering: Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering CASE Tools: What is CASE, Building Blocks of CASE, A Taxonomy of CASE Tools, Integrated CASE Environments, The Integration Architecture, The CASE Repository.

(12 Lectures Hours)

Text and Reference Books:

(1) Software Engineering, R. Pressman, McGraw-Hill.
(2) Software Engineering, K.K. Agrawal and Y. Sing, New Age International.
(3) Software Project Management in Practice, P. Jalote, Pearson.

Suggested E-resources:-

(1) https://www.cs.uct.ac.za/mit_notes/software/pdfs/SE_top.pdf
(2) https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf
(3) https://www.vssut.ac.in/lecture_notes/lecture1428551142.pdf

Semester III

Course Code : MIT9007T
Title of the Course : Compiler Design
Qualification Level of the Course: NHEQF Level 6.5
Credit of the Course : 4
Type of the Course: Discipline Centric Compulsory (DCC) Course in Computer Science
Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites:

1. Requires basic understanding of at least one programming language such as C, Java etc.

Objectives of the Course:

1. The objective of the compiler course is to understand the basic principles of compiler design, its various constituent parts, algorithms and data structures required to be used in the compiler.

Learning Outcomes:

1. To acquire basic skills for designing the compilers, as well as the knowledge of compiler design.
2. Able to understand the working of a compiler.
3. Understand the structure of a compiler, and how the source and target languages influence various choices in its design.
4. Give a new appreciation for programming language features and the implementation challenge
5. Understanding the details of typical compilation models
6. Understand some specific components of compiler technology, such as lexical analysis, grammars and parsing, type-checking, intermediate representations, etc

Syllabus

UNIT-I

Introduction to translators, compilers, interpreters, compilation process. Programming language grammars, derivations, reductions, regular expression, context free language and grammar. Lexical analyzer, input buffering, specification and recognition of tokens, introduction to finite automata, regular expressions to NFA, minimization of DFA, keywords and reserved word policies, LEX, the lexical analyzer generator.

(12 Lectures Hours)

UNIT-II

Syntax analyzer, context free grammars, top down parsing, brute force parser, recursive descent parser, LL(1) parser, Bottom up parsing, operator precedence parsing, simple precedence parsing, LR parser, LALR parser, YACC, the parser generator.

(12 Lectures Hours)

UNIT-III

Syntax directed translation schemes, implementation of syntax directed translators, synthesized attributes, inherited attributes, dependency graph, evaluation order, construction of syntax trees, directed acyclic graph of expression, bottom up evaluation of S- attributed definitions, L attributed definitions, top down translation of L - attributed definitions.

(12 Lectures Hours)

UNIT-IV

Errors, lexical phase errors, syntactic phase errors. Intermediate languages, postfix notation, syntax trees, parse trees, three address code, triples and indirect triples. Translation of assignment statements, Symbol tables, operation on symbol tables, and symbol table organization for non-block structured languages, symbol table organization for block, structured languages.

(12 Lectures Hours)

UNIT-V

Run time storage management, storage allocation and referencing data in block structured language, storage allocation. Code optimization, sources of optimization, loop optimization, DAG and optimization of basic blocks. Code generation, a machine model, next use information register allocation and assignment, a simple code generator, code generation from DAG's, Peephole optimization.

(12 Lectures Hours)

Text and Reference Books:

1. Compilers, Principles, techniques and tools, Aho, Ullman and Sethi, Pearson Education.
2. The Theory and Practice of Compiler Writing, Tremblay, Sorenson, BSP.
3. Compiler Design in C, Holub, PHI.

Suggested E-resources:-

(1) https://www.britannica.com/technology/compiler
(2) https://www.tutorialspoint.com/compiler_design/compiler_design_syntax_analysis.htm
(3) https://www.geeksforgeeks.org/error-handling-compiler-design/

DSE- Group A

Semester III

Course Code : MIT9104T
Title of the Course : Java Programming
Qualification Level of the Course : NHEQF Level 6.5
Credit of the Course : 4
Type of the Course: Discipline Specific Elective (DSE) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite:

(1) The students should have basic knowledge of programming languages like C and C++.
(2) Prior knowledge of Object-Oriented concepts is helpful but not mandatory.

Objectives of the Course : The learning objectives of this course are:

(1) To understand the basic concepts and fundamentals of platform independent object oriented language Java.
(2) To understand the concepts of packages and interfaces in java.
(3) To demonstrate skills in writing programs using exception handling techniques and multithreading.
(4) To understand the connectivity of the Java programming language with databases.
(5) To understand the advanced concepts of java including Servlet and Java Server Pages (JSP) used for web application development.

Learning Outcomes: After successful completion of the course, the students are able to:

(1) Use the syntax and semantics of java programming language and basic concepts of OOP.
(2) Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.
(3) Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.
(4) Design web related applications which mimic the real world scenarios using JDBC,Servlet and JSP.

Syllabus

UNIT-I

OOPS Concepts: Encapsulation, Inheritance and polymorphism, Classes and data abstraction, constructors and destructors. (12 Lectures Hours)

UNIT-II

Packages and Interfaces, Access Control, Method Overriding, Garbage Collection, Abstract Classes. Exceptions handling, throwing an exception, try block, catching an exception, Multithreading, Synchronization.

(12 Lectures Hours)

UNIT-III

J2EE Platform: Enterprise architecture style (2 tier, 3 tier, N tier), J2EE run time, J2EE APIs, J2EE technology, web components JDBC Overview, JDBC implementation, Connection class, Statements, Types of statement objects (Statement, Prepared Statement and Callable Statement), and Types of result set.

(12 Lectures Hours)

UNIT-IV

Servlet: Servlet API, Overview of Servlet, Servlet Life Cycle, HTTP Methods, Attributes in Servlet, Request Dispatcher interface.

(12 Lectures Hours)

UNIT-V

Java Server Pages: JSP Overview, Problem with Servlet, Life Cycle of JSP Page, JSP Processing, JSP Application Design with MVC, JSP Directives, JSP Action, JSP Implicit objects, JSP Session and Cookies Handling.

(12 Lectures Hours)

Text and Reference Books:

(1) Professional Java Server Programming by Subrahmanyam Allamaraju, Cedric Buest Wiley Publication
(2) Spring in Action 3rd edition , Craig walls, Manning Publication
(3) Hibernate 2nd edition, Jeff Linwood and Dave Minter, Beginning Après publication
(4) Java Server Faces in Action, Kito D. Mann, Manning Publication
(5) JDBC™ API Tutorial and Reference, Third Edition, Maydene Fisher, Jon Ellis, Jonathan Bruce, Addison Wesley
(6) Beginning JSP, JSF and Tomcat, Giulio Zambon, Apress
(7) Complete Reference J2EE by James Keogh McGraw publication

Suggested E-resources:

(1) https://ocw.mit.edu/courses/6-092-introduction-to-programming-in-java-january-iap-2010/pages/lecture-notes/
(2) https://math.hws.edu/javanotes/
(3) https://www.youtube.com/watch?v=hBh_CC5y8-s
(4) https://www.youtube.com/watch?v=BGTx91t8q50
(5) https://www.youtube.com/watch?v=BGTx91t8q50
(6) https://www.youtube.com/watch?v=eIrMbAQSU34
(7) https://www.youtube.com/watch?v=UmnCZ7-9yDY

DSE- Group A

Semester III

Course Code : MIT9105T
Title of the Course **:** Data Mining and Data Warehousing

Qualification Level of the Course : NHEQF Level 6.5

Credit of the Course : 4

Type of the Course: Discipline Specific Elective (DSE) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites:
Familiarize yourself with the fundamental concepts of databases, including data models (such as relational, hierarchical, and object-oriented), database design principles, and query languages (such as SQL). Understanding how data is organized and stored will be crucial for data mining and data warehousing.

Objectives of the course:
The course aims to provide an understanding of the principles and components of data warehousing. This involves learning about data integration, data cleaning, data transformation, and building data warehouses for efficient data storage and retrieval.

Learning Outcomes

(1) Learn Database Management System Concepts and Architecture and data warehouse
(2) Able to understand the Warehouse Implementation
(3) Able to Understand Data Preprocessing
(4) Learn Data Mining Methods
(5) Able to understand Fuzzy Logic Clustering and Introduction to Fuzzy Logic

Syllabus

UNIT I

Operational Database Systems and Data Warehouses, a multidimensional Data Model, Data Warehouse Architecture, Three-tier Data Warehouse Architecture, Steps for the design and construction of Data Warehouses

(12 Lectures Hours)

UNIT II

Conceptual Data Architecture, Logical Architectures-star schema, snowflake, constellation, facttable and dimensions tables, Design Techniques. Data Warehouse Implementation Data Warehouse and OLAP Technology for Data Mining, Data Warehouse and OLAP Technology for Data Mining:, Data Marts, Metadata, OLAP, Categorization of OLAP Tools.

(12 Lectures Hours)

UNIT III

Data Preprocessing (ETL Process): Data Cleaning, Data Integration and Transformation, Data Reduction, Data Mining Primitives, Concept Description, Mining Association Rules and Algorithms.

(12 Lectures Hours)

UNIT IV

Data Mining Methods Data Mining Methods –Correlation Analysis, Classification and Prediction - Basic Concepts, Statistical based classification, Decision Tree Induction, K Nearest Neighbors, Rule Based Classification, Classification by Backpropagation, Support Vector Machines (12 Lectures Hours)

UNIT V

Clustering and Introduction to Fuzzy Logic Clustering and Introduction to Fuzzy Logic: Cluster Analysis, Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Outlier Analysis, Data Mining Applications. Crisp set and Fuzzy set, Basic concepts of fuzzy sets, membership functions. Basic operations on fuzzy sets, Properties of fuzzy sets, Fuzzy relations. Propositional logic and Predicate logic (12 Lectures Hours)

Text and Reference Books:

1. Fundamentals of Database Systems, Elmasri, Navathe: Addison Wesley, Pearson Education.
2. Data Warehousing, Data Mining & OLAP, Alex Berson and Stephen J. Smith, Tata McGraw,Hill
3. Data Mining Concepts and Techniques, Jiawei Han and Micheline Kamber, Elsevier.
4. Introduction To Data Mining, Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Person Education.
5. Insight into Data mining Theory and Practice, K.P. Soman, ShyamDiwakar and V. Ajay, Prentice Hall of India.
6. Introduction to Data Mining with Case Studies, G. K. Gupta,Prentice Hall of India

Suggested E-resources:-

1. https://ia800702.us.archive.org/7/items/datamining_201811/DS-book%20u5.pdf
2. https://textbooks.elsevier.com/manualsprotectedtextbooks/9780123814791/Instructor's_manual.pdf
3. https://dl.ebooksworld.ir/motoman/Cambridge.University.Press.Data.Mining.and.Data.Warehousing.www.EBooksWorld.ir.pdf

DSE Group –B

Semester III

Course Code : MIT9106T
Title of the Course : Cloud Computing
Qualification Level of the Course : NHEQF Level 6.5
Credit of the Course : 4
Type of the Course : Discipline Specific Elective (DSE) Course in Computer Science
Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites : None
Objectives of the Course:

1. To provide students with the fundamentals and essentials of Cloud Computing.
2. To provide students a sound foundation of the Cloud Computing so that they are able to start using and adopting Cloud Computing services and tools in their real life scenarios.
3. To enable students exploring some important cloud computing driven commercial systems and applications.
4. To expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

Learning Outcomes:

After successful completion of this course, student will be able to

1. Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about.
2. Characteristics, advantages and challenges brought about by the various models and services in cloud computing.
3. Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.
4. Identify resource management fundamentals, i.e. resource abstraction, sharing and sandboxing and outline their role in managing infrastructure in cloud computing.
5. Analyze various cloud programming models and apply them to solve problems on the cloud.

Syllabus

UNIT I

Introduction Cloud Computing: Definition, Types of Clouds, Layer & Services models, Deployment models, Cloud Computing Architecture & infrastructure: Cloud Reference Model, Virtualization: Definition, Types of virtualizations (Compute, Network, Storage), Types of Hypervisor

(12 Lectures Hours)

UNIT II

Cloud Platforms in Industry: Major vendors and their offerings, Introduction to Microsoft Azure, Amazon web services (EC2, S3, Etc.), Google AppEngine. Integration of Private and Public Clouds

Cloud applications: Protein structure prediction, Data Analysis, Satellite Image Processing, CRM and ERP, Social networking. Cloud Application- Scientific Application, Business Application.

(12 Lectures Hours)

UNIT III

Advance Topic in Cloud Computing: Cloud Security, Risks and Approaches of Migration into Cloud. Federated Cloud/ Intercloud, Third Party Cloud Services, Business Continuity and Disaster Recovery, Service Level Agreement (SLA), Dynamic resource provisioning and management, Server consolidation and placement policies, Energy efficiency in data centers, Elastic Load Balancing and Auto Scaling.

(12 Lectures Hours)

UNIT IV

Storage Network Design: Architecture of storage, analysis and planning. Storage network design considerations; NAS and FC SANs, hybrid storage networking technologies (iSCSI, FCIP, FCoE), design for storage virtualization in cloud computing, host system design considerations. Techniques for Big data processing (Google GFS, BigTable, and Map-Reduce Hadoop Distributed File System (HDFS), HIVE).

(12 Lectures Hours)

UNIT V

Consensus in Cloud Computing: Issues in consensus, Consensus in synchronous and asynchronous system, Byzantine Agreement: Agreement, Faults, Tolerance, Measuring Reliability and Performance, SLIs, SLOs, SLAs, TLAs, Byzantine failure, Byzantine Generals Problem, Failures & Recovery Approaches in Distributed Systems,

(12 Lectures Hours)

Text and Reference Books:

(1) Distributed and Cloud Computing, Kai Hawang, GeofreyC.Fox, Jack J. Dongarra Elservier
(2) Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
(3) Cloud Computing, Kumar Saurabh, Wiley Pub
(4) Cloud Security, Krutz, Vines, Wiley Pub
(5) Cloud Computing- A Practical Approach, Velte, TMH

Suggested E-resources:

(1) https://onlinecourses.nptel.ac.in/noc21_cs14/preview
(2) https://www.cs.cmu.edu/~msakr/15619-s16/15319_15619_s16_Syllabus.pdf

DSE Group - B

Semester III

Course Code **:** MIT9107T
Title of the Course **:** Swift Programming
Qualification Level of the Course : NHEQF Level 6.5
Credit of the Course : 4

Type of the Course : Discipline Specific Elective (DSE) Course in Computer Science

Delivery Type of the Course: Lecture, 40+20=60. The 40 lectures for content delivery and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisites:
Students should be familiar with programming concepts like variables, if statements, and loops, as well as object-oriented concepts like methods and classes. You will also need access to a Mac computer running OS X 10.10 or later.

Objectives of the Course:

(1) Demonstrate an understanding of the fundamentals of Swift, building modern mobile apps, iOS, Xcode, and
(2) other tools in the Xcode development environment.
(3) Demonstrate and understanding of how to handle and store data using clearly defined types.
(4) Demonstrate problem solving skills by developing and implementing algorithms to solve problems using
(5) operators.
(6) Demonstrate an understanding of strings, functions, structures, collections, loops, and different ways to
(7) work with the information that makes up an app.
(8) Design and implement simple classes.
(9) Demonstrate an understanding of structure design by implementing programs with functions, including parameter passing and value returning.
(10) Demonstrate understanding using Interface Builder and storyboards to build the user interface for apps with multiple views.

Learning Outcomes:

1. Write code that makes decisions about what lines of code should be executed
2. Create a basic iOS app to get familiar using Xcode.
3. Test and debug apps in a Mac, using the Simulator from Xcode.
4. Create visual interfaces using the Interface Builder from Xcode.
5. Create multiple scenes, views, and controls to build simple workflows.

Syllabus

Unit I

Getting Started with App Development: Introduction to Swift and Playgrounds, Constants, Variables, and Data Types, Operators, Control Flow ,Building, Running, Debugging and App

(12 Lectures Hours)

Unit II

Introduction to UIKit, Strings, Functions, Collections, Loops, Interface builder basics

(12 Lectures Hours)

Unit III

Advanced concepts of UIKit, Structures, Classes, Inheritance, Introduction to UIKit, Display Data, Controls in Action, Auto Layout and Stack Views, Guided Project – Apple Pie

(12 Lectures Hours)

Unit IV

Navigation and Workflow, Optionals, Types Casting and Inspection, Guard, Constant and Variables Scope,Enumerations, Segues and Navigation Controllers, Tab Bar Controllers

(12 Lectures Hours)

Unit V

View Controllers Life Cycle, Building Simple Workflow, Guided Project – Apple pie, traffics Light, Hobby Tutorial, Personality, Quizetc

(12 Lectures Hours)

Text and Reference Books:-

(1) Develop in Swift Fundamentals Xcode 13
(2) Swift Programming: The Big Nerd Ranch Guide
(3) Hello Swift!: iOS app programming for kids and other beginners

Suggested E-resources:-

(1) https://developer.apple.com/swift/resources/
(2) https://docs.swift.org/swift-book/
(3) https://medium.com/level-up-web/best-swift-books-in-2017-e2b4d562825f

DSE Group –C

Semester III

Code of the Course : MIT8104P
Title of the Course : Java Programming Lab
Qualification Level of the Course : NHEQF Level 6.5
Credit of the Course : 4
Type of the Course : Discipline Specific Elective (DSE) Course in Computer Science

Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving..

Prerequisite:

(1) The students should have basic knowledge of programming languages like C.
(2) Prior knowledge of Object-Oriented Language like C++ is helpful but not mandatory.

Objectives of the course:

(1) The course aims to strengthen the conceptual knowledge of Object Oriented Programming Language JAVA at core level and advance level.
(2) lay foundation for further learning of the subject C++ and Advance Java Programming Language which is useful for the design of desktop and web applications in Computer Science

Learning Outcomes:

(1) The student will be able to learn conceptual and advanced level of programming with a modern programming language, Java.
(2) The student gets an understanding of how to install and use a good Java development environment.
(3) The student learn how to produce robust programs in Java using :
    - Variables
    - Expressions
    - Looping
    - Branching
    - Methods / Functions
    - Classes, Interfaces and Inheritance
    - Package
    - Exceptions
    - Multithreading

(4) The student will be able to learn connectivity of the Java programming language with databases using JDBC.
(5) Design web related applications which mimic the real world scenarios using servlets and JSP.

List of Some Programs for reference:

(1) Program to develop an in depth understanding of Java: data types, variables, operators, operator precedence,
(2) Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
(3) Write object oriented programs in Java: objects, classes, constructors and destructors.
(4) Write object oriented programs of inheritance, using super, final with inheritance, overloading and overriding
(5) Programs on access control,static member, abstract classes and variables.
(6) Program to develop understanding of packages & Interfaces in Java
(7) Write a program using exception handling mechanisms.
(8) Programs using JDBC.
(9) Programs using Servlets.
(10) Programs using JSP for web applications.

Text and Reference Books:

(1) Professional Java Server Programming by Subrahmanyam Allamaraju, Cedric Buest Wiley Publication
(2) Spring in Action 3rd edition , Craig walls, Manning Publication
(3) Hibernate 2nd edition, Jeff Linwood and Dave Minter, Beginning Après publication
(4) Java Server Faces in Action, Kito D. Mann, Manning Publication
(5) JDBC™ API Tutorial and Reference, Third Edition, Maydene Fisher, Jon Ellis, Jonathan Bruce, Addison Wesley
(6) Beginning JSP, JSF and Tomcat, Giulio Zambon, Apress
(7) Complete Reference J2EE by James Keogh McGraw publication

Suggested E-resources:

(1) https://ocw.mit.edu/courses/6-092-introduction-to-programming-in-java-january-iap-2010/pages/lecture-notes/
(2) https://math.hws.edu/javanotes/
(3) https://www.youtube.com/watch?v=hBh_CC5y8-s
(4) https://www.youtube.com/watch?v=BGTx91t8q50
(5) https://www.youtube.com/watch?v=BGTx91t8q50
(6) https://www.youtube.com/watch?v=eIrMbAQSU34
(7) https://www.youtube.com/watch?v=UmnCZ7-9yDY

Scheme of EoSE: -

1) Exercise(s): 45 2) Viva Voce: 20 3) Evaluation of record book: 15

DSE Group –C

Semester III

Code of the Course : MIT8105P
Title of the Course : Data Mining and Data Warehousing Lab
Qualification Level of the Course : NHEQF Level 6.5
Credit of the Course : 4
Type of the Course : Discipline Specific Elective (DSE) Course in Computer Science

Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving..

Prerequisites: Data Modelling, DBMS or equivalent.

Objectives of the Course:

Learn how to perform data mining tasks using a data mining toolkit (such as open source WEKA), Understand the data sets and data pre-processing, Demonstrate the working of algorithms for data mining tasks such as association rule mining, classification, clustering and regression, Exercise the data mining techniques with varied input values for different parameters. This course helps the students to practically understand a data warehouse, techniques and methods for data gathering and data pre-processing using OLAP tools. The different data mining models and techniques will be discussed in this course.

(1) To teach principles, concepts and applications of data warehousing and data mining
(2) To introduce the task of data mining as an important phase of knowledge recovery process
(3) To inculcate Conceptual, Logical, and Physical design of Data Warehouses OLAP applications and OLAP deployment
(4) To inculcate fundamental concepts that provide the foundation of data mining
(5) Design a data warehouse or data mart to present information needed by management in a form that is usable for management client
(6) Ability to understand the various kinds of tools.
(7) Demonstrate the classification clusters and etc. in large data sets

List of sample programs for reference

1. Explore WEKA Data Mining/Machine Learning Toolkit
    a. Downloading and/or installation of WEKA data mining toolkit,
    b. Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
    c. Navigate the options available in the WEKA
    d. (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel)
    e. Study the arff file format
    f. Explore the available data sets in WEKA.

2. Load a data set (ex. Weather dataset, Iris dataset, etc.) Load each dataset and observe the following:
    a. List the attribute names and they types
    b. Number of records in each dataset
    c. Identify the class attribute (if any)
    d. Plot Histogram
    e. Determine the number of records for each class.
    f. Visualize the data in various dimensions
3. Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets
    a. Explore various options available in Weka for preprocessing data and apply unsupervised filters like Discretization, Resample filter, etc. on each dataset
    b. Load weather. nominal, Iris, Glass datasets into Weka and run Apriori algorithm with different support and confidence values. Study the rules generated.
    c. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated. Derive interesting insights and observe the effect of discretization in the rule generation process.
4. Demonstrate performing classification on data sets
    a. Load each dataset into Weka and run 1d3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.
    b. Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix.
    c. Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained.
    d. Plot RoC Curves
    e. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.
5. Demonstrate performing clustering of data sets
    a. Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
    b. Explore other clustering techniques available in Weka.
    c. Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain.
6. Demonstrate knowledge flow application on data sets
    a. Develop a knowledge flow layout for finding strong association rules by using Apriori,FP- Growth algorithms
    b. Set up the knowledge flow to load an ARFF (batch mode) and perform a crossvalidation using
    c. J48 algorithm
    d. Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree

Text and Reference Books:

1. Dunham M H, "Data Mining: Introductory and Advanced Topics", Pearson Education, New Delhi, 2003. Estd.
2. Jaiwei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Elsevier, 2006.
3. M Sudeep Elayidom, "Data Mining and Warehousing", 1st Edition, 2015, Cengage Learning India Pvt. Ltd.

2014

4. Mehmed Kantardzic, "Data Mining Concepts, Methods and Algorithms", John Wiley and Sons, USA, 2003.
5. Pang-Ning Tan and Michael Steinbach, "Introduction to Data Mining", Addison Wesley, 2006.

Suggested E-Resources:

1. Online virtual lab
2. https://www.cs.waikato.ac.nz/ml/weka https://weka.wikispaces.com

Scheme of EoSE :

(i) Exercise(s):45 (ii) Viva Voce: 20 (iii) Evaluation of record book: 15

GEC Group - B

Semester III

Course Code : MIT9106P
Title of the Course : Cloud Computing Lab

Qualification Level of the Course : NHEQF Level 6.5

Credit of the Course : 4

Type of the Course : Generic Elective Course (GEC) in Computer Science

Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.

Prerequisite **:None**

Objectives of the Course:

(1) To provide students with the fundamentals and essentials of Cloud Computing.
(2) To provide students a sound foundation of the Cloud Computing so that they are able to start using and adopting Cloud Computing services and tools in their real life scenarios.
(3) To enable students exploring some important cloud computing driven commercial systems and applications.
(4) To expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

Learning Outcomes:After successful completion of this course, student will be ableto

(1) Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about.
(2) Characteristics, advantages and challenges brought about by the various models and services in cloud computing.
(3) Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.
(4) Identify resource management fundamentals, i.e. resource abstraction, sharing and sandboxing and outline their role in managing infrastructure in cloud computing.
(5) Analyze various cloud programming models and apply them to solve problems on the cloud.

List of Experiments

(1) Use gcc to compile c-programs. Split the programs to different modules and create an application using make command.

(2) Use version control systems command to clone, commit, push, fetch, pull, checkout, reset, and delete repositories.
(3) Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
(4) Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
(5) Install Google App Engine. Create hello world app and other simple web applications using python/java.
(6) Use GAE launcher to launch the web applications.
(7) Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
(8) Find a procedure to transfer the files from one virtual machine to another virtual machine.
(9) Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version) Install Hadoop single node cluster and run simple applications like word count

Text and Reference Books:

(1) Distributed and Cloud Computing, Kai Hawang, GeofreyC.Fox, Jack J. DongarraElservier
(2) Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
(3) Cloud Computing, Kumar Saurabh, Wiley Pub
(4) Cloud Security, Krutz, Vines, Wiley Pub
(5) Cloud Computing- A Practical Approach, Velte, TMH

Suggested E-resources:

(1) https://onlinecourses.nptel.ac.in/noc21_cs14/preview
(2) https://www.youtube.com/watch?v=HhStJ7FDBpc&ab_channel=5MinutesEngineering
(3) https://www.cs.cmu.edu/~msakr/15619-s16/15319_15619_s16_Syllabus.pdf

Scheme of EoSE: - 1) Exercise(s): 45
2) Viva Voce: 20
3) Evaluation of record book: 15

GEC Group - B

Semester III

Course Code : MIT9107P
Title of the Course : Swift Programming Lab

Qualification Level of the Course : NHEQF Level 6.5

Credit of the Course : 4
Type of the Course : Generic Elective Course (GEC) in Computer Science

Delivery Type of the Course: Practicum 80+40=120. The 80 hours for Lab and 20 hours for lab practices and 20 hours on diagnostic assessment, formative assessment, and subject/class activity, problem solving.Prerequisite: Students should be familiar with programming concepts like variables, if statements, and loops, as well as object-oriented concepts like methods and classes. You will also need access to a Mac computer running OS X 10.10 or later.

Objectives of the Course:

(1) Demonstrate an understanding of the fundamentals of Swift, building modern mobile apps, iOS, Xcode, and
(2) other tools in the Xcode development environment.
(3) Demonstrate and understanding of how to handle and store data using clearly defined types.
(4) Demonstrate problem solving skills by developing and implementing algorithms to solve problems using
(5) operators.
(6) Demonstrate an understanding of strings, functions, structures, collections, loops, and different ways to
(7) work with the information that makes up an app.
(8) Design and implement simple classes.
(9) Demonstrate an understanding of structure design by implementing programs with functions, includingparameter passing and value returning.
(10) Demonstrate understanding using Interface Builder and storyboards to build the user interface for appswith multiple views.

Learning Outcomes:

(1) Write code that makes decisions about what lines of code should be executed
(2) Create a basic iOS app to get familiar using Xcode.
(3) Test and debug apps in a Mac, using the Simulator from Xcode.
(4) Create visual interfaces using the Interface Builder from Xcode.
(5) Create multiple scenes, views, and controls to build simple workflows.

List of Experiments are based on

Getting Started with App Development

a. Introduction to Swift and Playgrounds
b. Constants, Variables, and Data Types
c. Operators
d. Control Flow
e. Excode
f. Building, Running, Debugging and App
g. Documentation
h. Interface Builder Basics
i. Guided Project – Light

Introduction to UIKit

a. Strings
b. Functions
c. Structures
d. Classes, Inheritance
e. Collections
f. Loops
g. Introduction to UIKit
h. Display Data
i. Controls in Action
j. Auto Layout and Stack Views
k. Guided Project – Apple Pie

Navigation and Workflow

a. Optionals
b. Types Casting and Inspection
c. Guard
d. Constant and Variables Scope
e. Enumerations
f. Segues and Navigation Controllers
g. Tab Bar Controllers
h. View Controllers Life Cycle
i. Building Simple Workflow
j. Guided Project – Personality Quiz

Text and Reference Books:-

(1) Develop in Swift Fundamentals Xcode 13
(2) Swift Programming: The Big Nerd Ranch Guide
(3) Hello Swift!: iOS app programming for kids and other beginners

Suggested E-resources:-

(1) https://developer.apple.com/swift/resources/
(2) https://docs.swift.org/swift-book/
(3) https://medium.com/level-up-web/best-swift-books-in-2017-e2b4d562825f

Scheme of EoSE: -

1) Exercise(s): 45

2) Viva Voce: 20

3) Evaluation of record book: 15

Semester IV

Code of the Course : MIT9008T
Title of the Course : Report Writing
Qualification Level of the Course : NHEQF Level 6.5
Credit of the Course : 2
Type of the Course : Discipline Specific Elective (DSE) Course in Computer Science

Details

Research/ Project report based on topic selected by student based on guidance of faculty/industry expert/any other mechanism of delivery adopted by department committee at the execution of course etc.

DSE Group –D

Semester IV

Code of the Course	: MIT91XXP
Title of the Course	: Project
Qualification Level of the Course	: NHEQF Level 6.5
Credit of the Course	: 18
Type of the Course	:Discipline Specific Elective (DSE) Course in Computer Science

Details

In house project must be done by each student on simple applications using any computer language/ RDBMS/ Web design/visual programming etc. The total work must be of minimum 540 hours per student. The internal guide must schedule the work & evaluate internally from time to time. The project report must be prepared for the external examination. Monthly report of the students must be taken to monitor progress and must be placed for evaluation by external examiner. Projects submitted by the students shall be evaluated during external evaluation to ensure independent contribution and proficiency acquired by the students. Note: Students must be allotted projects in the beginning of the session. Candidates submitting readymade projects/copied/ projects developed by professionals in the market etc shall be awarded zero marks. Two copies of the project report and the software developed must be submitted to the external examiner. One copy of the project shall be returned to the student with the signature of external examiner.