

**Department of Computer Science
Islamic University of Science & Technology**



**Curriculum
(Choice Based Credit System)**

for

**Master of Science in Information Technology
(M.Sc.IT)
Two Year Programme
2023 and Onwards**

CBCS Course Outline MSc.IT



List of Abbreviations

- *FC* “*Foundation Course*”
- *SEC* “*Skill Enhancement Course*”
- *VAC* “*Value Added Course*”
- *AEC* “*Ability Enhancement Course*”
- *MDC* “*Multidisciplinary Course*”
- *EC* “*Employability Course*”
- *EOC* “*Entrepreneurship Oriented Course*”



Course Outline MSc.IT Semester I

Semester-I (24 Credit Semester with 31 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT501C	Programming Concepts in C/C++	Core	5	3	0	4	07	SEC
MIT502C	Database Management System	Core	5	3	0	4	07	EC
MIT503C	Discrete Mathematics	Core	4	3	1	0	04	FC
MIT504C	Computer Organization and Architecture	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MIT511E	Digital Electronics	DCE	3	2	1	0	03	SEC
MIT512E	Green Technologies	DCE	3	2	1	0	03	VAC
MIT513E	Management Information System	DCE	3	2	1	0	03	EC
Pool B: 3 Discipline Centric Elective Credit Units								
MIT526E	Web Technologies	DCE	3	2	0	2	04	EOC
MIT527E	E-Commerce	DCE	3	2	0	2	04	EC
MIT528E	Multimedia Technologies	DCE	3	2	0	2	04	EC
Audit Courses (Mandatory)								
MIT541A	Information and Communications Technology	Audit	0	2	0	0	02	AEC
Total			24				31	

Course Outline MSc.IT Semester II

Semester-II (26 Credit Semester with 33 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT551C	Data Structures	Core	5	3	0	4	07	FC
MIT552C	Introduction to Python	Core	5	3	0	4	07	SEC
MIT553C	Data Communication & Computer Networks	Core	4	3	1	0	04	FC
MIT554C	Operating System	Core	4	3	1	0	04	FC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MIT561E	Web Mining	DCE	3	2	1	0	03	SEC
MIT562E	Quantitative Techniques	DCE	3	2	1	0	03	AEC
MIT563E	Microprocessor & Assembly Language Programming	DCE	3	2	1	0	03	AEC
MIT564E	Internet of Things	DCE	3	2	1	0	03	SEC
Pool B: 03 Discipline Centric Elective Credit Units								
MIT576E	Open Source Technologies	DCE	3	2	0	2	04	EOC
MIT577E	Computer Graphics	DCE	3	2	0	2	04	SEC
MIT578E	Numerical and Statistical Computing	DCE	3	2	0	2	04	VAC
MIT579E	Pervasive Computing	DCE	3	2	0	2	04	AEC
Audit Courses (Mandatory)								
MIT591A	Technical Communication	Audit	0	2	0	0	02	VAC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			26				33	

**** to be chosen from other offering Departments of IUST**

MSc. IT Syllabus – Department of Computer Science, IUST

Course Outline MSc.IT Semester III

Semester-III (26 Credit Semester with 31 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT601C	Full-Stack Development	Core	5	3	0	4	07	EC
MIT602C	Artificial Intelligence & Machine Learning	Core	5	3	0	4	07	AEC
MIT603C	Software Engineering	Core	4	3	1	0	04	FC
MIT604C	Design and Analysis of Algorithms	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MIT611E	Mobile Communication	DCE	3	2	1	0	03	AEC
MIT612E	Distributed Databases	DCE	3	2	1	0	03	EC
MIT613E	Cryptography and Network Security	DCE	3	2	1	0	03	SEC
MIT614E	System Software Design	DCE	3	2	1	0	03	AEC
MIT615E	Natural Language Processing	DCE	3	2	1	0	03	AEC
Pool B: 3 Discipline Centric Elective Credit Units								
MIT626E	Digital Image Processing	DCE	3	2	0	2	04	AEC
MIT627E	Object Oriented Analysis & Design	DCE	3	2	0	2	04	EC
MIT628kE	Data Warehouse and Mining	DCE	3	2	0	2	04	EC
MIT629E	Java Programming	DCE	3	2	0	2	04	EC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			26				31	

**** to be chosen from other offering Departments of IUST**

Course Outline MSc.IT Semester IV

Semester-IV (24 Credit Semester and 32 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT651C	Theory of Computation & Formal Languages	Core	4	3	1	0	04	FC
MIT652C	Big Data Analytics	Core	4	3	1	0	04	AEC
MIT653C	Project work	Core	10	1	2	7	17	EOC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MIT661E	Research Methodology	DCE	3	2	1	0	03	FC
MIT662E	Cloud and Grid Computing	DCE	3	2	1	0	03	SEC
MIT663E	Cyber Security	DCE	3	2	1	0	03	SEC
MIT664E	Compiler Design	DCE	3	2	1	0	03	AEC
MIT665E	Bioinformatics	DCE	3	2	1	0	03	AEC
MIT666	Generative AI	DCE	3	2	1	0	03	AE
Pool B: 03 Discipline Centric Elective Credit Units								
MIT676E	Block Chain Technologies	DCE	3	2	0	2	04	EC
MIT677E	Quantum Computing	DCE	3	2	0	2	04	AEC
MIT678E	Advanced Java	DCE	3	2	0	2	04	EC
MIT679E	Deep Learning	DCE	3	2	0	2	04	SEC
Total			24				32	



Semester - I

MSc. IT Syllabus – Department of Computer Science, IUST

Semester-I (24 Credit Semester with 31 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT501C	Programming Concepts in C/C++	Core	5	3	0	4	07	SEC
MIT502C	Database Management System	Core	5	3	0	4	07	EC
MIT503C	Discrete Mathematics	Core	4	3	1	0	04	FC
kMIT504C	Computer Organization and Architecture	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MIT511E	Digital Electronics	DCE	3	2	1	0	03	SEC
MIT512E	Green Technologies	DCE	3	2	1	0	03	VAC
MIT513E	Management Information System	DCE	3	2	1	0	03	EC
Pool B: 3 Discipline Centric Elective Credit Units								
MIT526E	Web Technologies	DCE	3	2	0	2	04	EOC
MIT527E	E-Commerce	DCE	3	2	0	2	04	EC
MIT528E	Multimedia Technologies	DCE	3	2	0	2	03	EC
Audit Courses (Mandatory)								
MIT541A	Information and Communications Technology	Audit	0	2	0	0	02	AEC
Total			24				31	

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Programming Concepts in C/C++
Semester: 1st
Credits: 05
Pre Requisite: -----

Course Code: MIT501C
Paper Type: Core
Max Marks: 125
Co-Requisite: MIT503C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs, applications in C. Also by learning the basic programming constructs they can easily switch over to any other language in future.*
- *Develop an in-depth understanding of functional, logic, and object-oriented programming paradigms.*
- *To practice the fundamental programming methodologies in the C/C++ programming language via laboratory experiences.*
- *To code, document, test, and implement a well-structured, robust computer program using the C/C++ programming language.*
- *Perform object oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O and other standard language constructs.*
- *Demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.*

COURSE CONTENT:

UNIT I

C programming language: Evolution, Features & Importance. Basic Structure of C programs, Character Set, Identifiers, Reserved Words, Data Types, Constants, Variables, Symbolic Constants, Casting and Standard Libraries. Logical and Control Structures: Assignment, Arithmetic, Relational, Logical, Compound, Increment, Decrement, Bitwise Operators & Special Operators. IF, IF – ELSE, Nested IF – ELSE, ?: , SWITCH CASE. Looping Constructs: FOR, WHILE, DO-WHILE, EXIT, BREAK, CONTINUE

UNIT II

Arrays: Types of arrays, Initialization, dynamic arrays. Character Arrays & Strings. String-handling functions.

Functions: Concepts, Elements, Prototypes & Types. Passing Arrays to Functions. Storage classes, Recursion. Command-line arguments. Multifile programming. Preprocessing. Pointers: Concepts, Variables, swapping data, swapping address v/s data, pointers & arrays,

UNIT III

Pointers to pointers, pointer to strings, pointer arithmetic, additional operators, pointers to functions, void pointers.

Structures and Unions: Syntax & use, members, structures & pointers, array of structures, structures & functions, structure within structures.

OOPS: Evolution and need of C++, Advantages over Procedural programming. Introduction to classes and objects, Basic OOPS programming.

UNIT IV

C++ Functions: passing arguments to functions, returning values from functions, reference arguments, inline functions, default arguments, object as function argument, returning objects from functions. Constructors and Destructors, Copy Constructors

Inheritance and Polymorphism: Inheritance and types, Polymorphism (static and dynamic), function overloading, function overriding, virtual functions & operator overloading.

MSc. IT Syllabus – Department of Computer Science, IUST

COURSE OUTCOMES:

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

- Formulate simple algorithms for arithmetic and logical problems, translate the algorithms to programs (in C language),
- Test and execute the programs and correct syntax and logical errors,
- Develop efficient algorithms for solving a problem and implementation.
- Use the various constructs of a programming language viz. conditional, iteration and Recursion.

Text Books:

1. Programming in ANSI C 6th Edition “E. Balaguruswamy”
2. Robert Lafore, “Object Orientation with C++ Programming”, Waite Grou

References:

1. Object Oriented Programming with C++ “ E. Balagurusamy”
2. Herbert Schildt, “C++ The Complete Reference”, Tata McGraw Hill
3. Dennis Richie & Kernighan, “C Programming Language”, Prentice Hall
4. Dietel & Dietel, “How to program”, Pearson Education

List of Practicals:

LAB JOURNAL_MIT501C	
S. No.	Program Title
1.	WAP to perform various Arithmetic Operations like: i. Addition of two numbers. ii. Subtraction of two numbers. iii. Multiplication of two numbers. iv. Simple division of two numbers. v. Modular division of two numbers.
2.	WAP to check whether a number is even or odd.
3.	WAP to show the use of typedef.
4.	WAP to show the use of enum.
5.	WAP to show the use of Relational Operators (<, >, <=, >=, ==, !=)
6.	WAP to show the use of logical Operators like: i. Logical AND (&&) ii. Logical OR (). iii. Logical NOT (!).
7.	WAP to show the use of Pre / Post Increment.
8.	WAP to show the use of Pre / Post Decrement.
9.	WAP to show the use of Ternary /Conditional operator.
10.	WAP to show the use of sizeof operator.
11.	WAP to show the use of Bitwise operators as: i. Bitwise OR. ii. Bitwise NOT. iii. Bitwise <<. iv. Bitwise >>.

MSc. IT Syllabus – Department of Computer Science, IUST

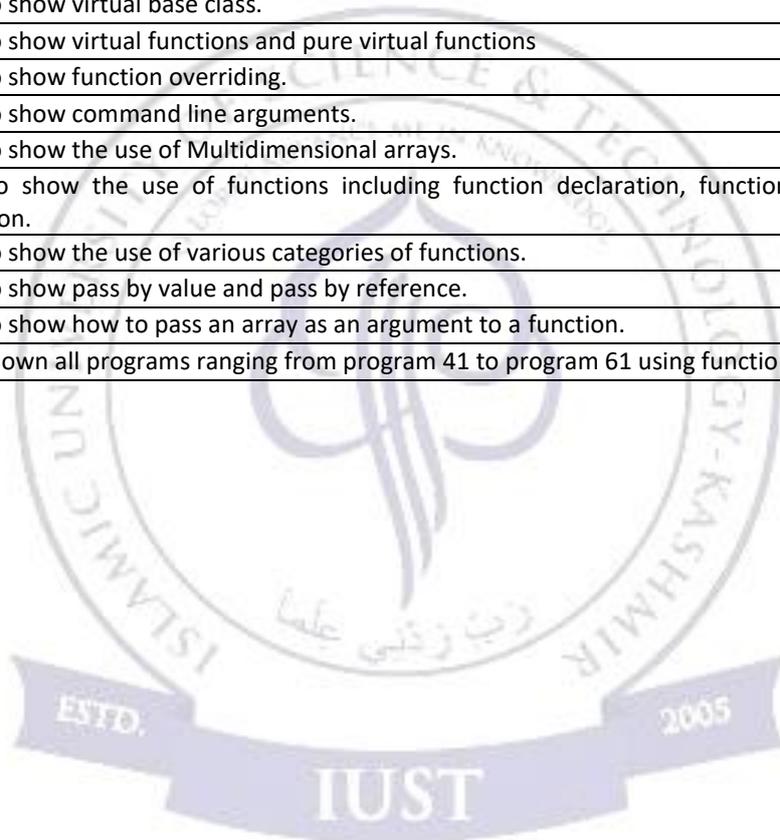
12.	WAP to show the user of Implicit type conversion.
13.	WAP to show the use of Explicit type conversion.
14.	WAP to show the use of simple if (Greater among two numbers).
15.	WAP to show the use of simple if (Greater among any numbers).
16.	WAP to show the use of if-else (Salary increase if Salary>10000).
17.	WAP to show the use of else-if (Grading of Students).
18.	WAP to show the use of switch (Months of Year).
19.	WAP to show the use of switch (Days of Week).
20.	WAP to show the use of while loop (Sum of digits of a number).
21.	WAP to show the use of while loop (Reverse of a number).
22.	WAP to show the use of while loop (X raised power n).
23.	WAP to print Fibonacci series using while loop.
24.	WAP to print whether number is Prime or not using while loop.
25.	WAP to print table using while loop.
26.	WAP to print sum of odd series using while loop.
27.	WAP to print sum of even series using while loop.
28.	WAP to find factorial of any number using while loop.
29.	WAP to show the use of for loop (Sum of digits of a number).
30.	WAP to show the use of for loop (Reverse of a number).
31.	WAP to show the use of for loop (X raised to Power n).
32.	WAP to print Fibonacci series using for loop.
33.	WAP to print whether number is Prime or not using for loop.
34.	WAP to print table using for loop.
35.	WAP to print sum of odd series using for loop.
36.	WAP to print sum of even series using for loop.
37.	WAP to find factorial of any number using for loop
38.	WAP for calculating the sum of even /odd series upto 'n'.
39.	WAP for converting a given number of days into months and remaining days.
40.	WAP to draw different structures like Triangle, Pyramid etc.
41.	WAP for calculating the sum of elements of array.
42.	WAP for obtaining the greatest element of array.
43.	WAP to insert an element into an array at different location.
44.	WAP for displaying an entered string.
45.	WAP to find frequency of elements in an array of 'n' elements.
46.	WAP for calculating string length. <ul style="list-style-type: none"> • Including spaces. • Excluding spaces.
47.	WAP to reverse the elements of an array of 'n' elements.
48.	WAP to delete an element in an array at a given location. <ul style="list-style-type: none"> • At start. • At some specific given position. • At the end of array.
49.	WAP for Addition of two Matrices.
50.	WAP for Subtraction of two Matrices.
51.	WAP for Multiplication of two Matrices.
52.	WAP for obtaining the transpose of a Matrix.
53.	WAP to implement a stack structure with its operations using arrays.
54.	WAP to implement a Queue structure with its consequent operations using arrays.

MSc. IT Syllabus – Department of Computer Science, IUST

55.	WAP to find length of a string without using any string handling function.
56.	WAP to perform string concatenation, string comparison without using any string handling function.
57.	WAP to perform string concatenation, string comparison while using string handling functions.
58.	WAP to find length of a string using string handling function.
59.	WAP to search a particular word from a given string.
60.	WAP to show the use of two dimensional arrays.
61.	WAP to show the use of Multidimensional arrays.
62.	WAP to show the use of functions including function declaration, function call and function definition.
63.	WAP to show the use of various categories of functions.
64.	WAP to show pass by value and pass by reference.
65.	WAP to show how to pass an array as an argument to a function.
66.	Write down all programs ranging from program 41 to program 61 using functions.
67.	WAP to show recursion (Factorial of a number).
68.	WAP to show the use of Storage classes.
69.	WAP to show Multifile programming.
70.	WAP to show the use of Pointers.
71.	WAP to show usage of Structures.
72.	WAP to show Copying and Comparing Structures.
73.	WAP to show the use of Array of Structure variables.
74.	WAP to show the use of Arrays within Structures.
75.	WAP to show the use of Structures within structures.
76.	WAP to show Structures and Functions.
77.	WAP to show the use of Unions.
78.	WAP to show the use of Bit fields.
79.	WAP to show the use and understanding of Pointers (Declaring, Initializing and Accessing pointers).
80.	WAP to show the use and understanding of chain of Pointers.
81.	WAP to show the use of Pointer Expressions.
82.	WAP to show the use of Pointer Increments / Decrements and Scale Factor.
83.	WAP to show the use of Pointers and Arrays.
84.	WAP to show the use of Pointers and Character arrays.
85.	WAP to show the use of Array of Pointers.
86.	WAP to show the use of Pointers as Function arguments.
87.	WAP to show the use of Functions returning Pointers.
88.	WAP to show the use of Pointers and Structures.
89.	WAP to show the use of Classes.
90.	WAP to show the use of Functions in C++ (Function prototype)
91.	WAP on Inline Functions.
92.	WAP to use call by reference and return by reference.
93.	WAP on default arguments.
94.	WAP to show Function Overloading.
95.	WAP in C++ to show nesting of functions.
96.	WAP to show static data members and static member functions.
97.	WAP to show objects as function arguments and returning objects.
98.	WAP on default and reference arguments
99.	WAP to show array of objects, their declaration and their passing to a function.
100.	WAP on constructors.

MSc. IT Syllabus – Department of Computer Science, IUST

101.	WAP on parameterized constructors.
102.	WAP on multiple constructors in a class.
103.	WAP on copy constructor and its use.
104.	WAP on destructors.
105.	WAP to show Inheritance in general.
106.	WAP to show single Inheritance.
107.	WAP to show Multiple Inheritance.
108.	WAP to show Multilevel Inheritance.
109.	WAP to show Hierarchical Inheritance.
110.	WAP to show Hybrid Inheritance.
111.	WAP on Operator Overloading.
112.	WAP to show this pointer.
113.	WAP to show virtual base class.
114.	WAP to show virtual functions and pure virtual functions
115.	WAP to show function overriding.
116.	WAP to show command line arguments.
117.	WAP to show the use of Multidimensional arrays.
118.	WAP to show the use of functions including function declaration, function call and function definition.
119.	WAP to show the use of various categories of functions.
120.	WAP to show pass by value and pass by reference.
121.	WAP to show how to pass an array as an argument to a function.
122.	Write down all programs ranging from program 41 to program 61 using functions.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Database Management System	Course Code:	MIT502C
Semester:	1 st	Paper Type:	Core
Credits:	05	Max Marks:	125
Pre Requisite:	-----	Co-Requisite:	MIT513E
Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To understand the role of a database management system in an Organization.*
- *To understand basic database concepts including the structure and Operation of the relational data model.*
- *To construct simple and moderately advanced database queries using Structured Query Language (SQL).*
- *To understand and successfully apply logical database design principles, including E-R diagrams and database normalization.*

COURSE CONTENT:

UNIT I

Basic Concepts and Conceptual Database Design: Database Users, Characteristics of the Database, Advantage of using Database Systems, Data Models, schemas and instances, Three Tier Architecture & Data Independence, Database Languages & Interfaces. Overview of Legacy Data Base Management Systems. Data Modeling Using The Entity-Relationship Model – Entities, Attributes and Relationships, Cardinality of Relationships, Strong and Weak Entity Sets, Translating your ER Model into Relational Model.

UNIT II

Relational Model, Languages & Systems: Relational Data Model, Relational Model Concepts, Relational Model Constraints, Relational Algebra, SQL – A Relational Database Language, Data Definition & Manipulation in SQL, Queries in SQL, Specifying Constraints in SQL, Practicing SQL commands using ORACLE, “PL/SQL, Stored Procedures”.

UNIT III

Functional Dependencies & Normalization for Relational Databases: Functional Dependencies, Normal Forms based on primary keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependencies.

“Mini Project: Data Analysis and Data Modelling”

UNIT IV

Transaction Management: Transaction Concept and State, Desirable Properties of a Transaction, Characterizing Schedules based on Serializability and Recoverability, Concurrency Control Techniques: Lock-Based Protocols, Timestamp-based Protocols, Validation-based Protocols. Database Recovery Techniques: Recovery Concepts, Recovery based on Deferred Update and Immediate Update. Shadow Paging. Overview of Object Oriented Database Management Systems, Distributed Data Base Management Systems.

COURSE OUTCOMES:

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

- *Introduces the role of a database management system, basic database concepts, including the structure and operation of the relational data model.*
- *Familiarize themselves with the concepts of integrity constraints, relational algebra, relational domain & tuple calculus, data normalization.*
- *Construct simple and moderately advanced database queries using Structured Query Language (SQL).*
- *Have knowledge of database transaction including concurrency control, backup and recovery, and data object locking.*
- *Design and implementation of a small database project using Oracle.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. Korth, Silberschatz, "Database System Concepts", TMH
2. Elmsari & Navathe, "Fundamentals of Database Systmes", A. Wesley
3. Ullman J. D., "Principles of Database Systems", Galgotia Publications

References:

1. Steve Bobrowski, "Oracle 8 Architecture", TMH
2. Date C. J., "An Introduction to Database Systems", Narosa Publishing
3. William Page, "Using Oracle 8i – Special Edition", Que/PHI
4. Ivan Bayross, "SQL & PL/SQL Using Oracle 8i & 9i with SQLJ", BPB
5. Desai.B, "An introduction to Database Concepts", Galgotia Publications

List of Practicals:

LAB JOURNAL_MIT502C	
S. No.	Program Title
1.	Study of DBMS, RDBMS.
2.	To study Data Definition language Statements.
3.	To study Data Manipulation Statements.
4.	Study of SELECT command with different clauses.
5.	Study of SINGLE ROW functions (character, numeric, Data functions).
6.	Study of GROUP functions (avg, count, max, min, Sum).
7.	Study of various type of SET OPERATORS (Union, Intersect, Minus).
8.	Study of various type of Integrity Constraints.
9.	Study of Various type of JOINS. To create a DDL to perform creation of table, alter, modify and drop column.
10.	Define various Database Languages.
11.	Write a program to create a database.
12.	Write a program to create table with constraints such as 'NOT NULL', 'UNIQUE', 'DEFAULT', 'PRIMARY KEY', 'CHECK' etc.
13.	Write a program to add a few record to the database.
14.	Write a program to insert values into a specific column.
15.	Write a program to study the viewing commands (select, update) and executes the following queries: <ul style="list-style-type: none">• Find the names of all students who study in MIT 3rd year.• Find the student names whose age is greater than 20 years.
16.	Write a program for following query to modify the structure of table using Alter or Delete command: <ul style="list-style-type: none">• Add an attribute named 'Phone' to the table 'Student'.• Drop the attribute 'Gender' from the table 'Student'.• Delete the entries from the table 'Student' who left the course in between.
17.	WAP to Update the record using 'WHERE' clause and without using 'WHERE' clause.
18.	Write a program to update the record using Sub Query.
19.	Write a program to add and remove a 'FOREIGN KEY'.
20.	Write a program to order the records in ascending and descending order.
21.	Write a program to Group the records using 'HAVING' clause.

MSc. IT Syllabus – Department of Computer Science, IUST

22.	Write programs for Aggregate functions such as 'SUM', 'COUNT', 'AVG', 'MAX', 'MIN'.
23.	Write programs for various relational algebraic operations such as 'UNION', 'INTERSECT', 'EXCEPT'.
24.	Write a program for Join operation as 'INNER JOIN', 'OUTER JOIN', and 'CROSS JOIN'.
25.	<p>Consider the following relations for an order processing database application in a company.</p> <p>CUSTOMER (Cust #: int, Cname: string, City: string) ORDER (Order #: int, Odate: date, Cust #: int, Ord-Amt: int) ORDER-ITEM (Order #: int, Item #: int, qty: int) ITEM (Item #: int, Unit Price: int) SHIPMENT (Order #: int, Warehouse #: int, Ship-Date: date) WAREHOUSE (Warehouse #: int, City: string)</p> <p>i) Create the above tables by properly specifying the primary keys and the foreign keys. ii) Enter at least five tuples for each relation. iii) Produce a listing: CUSTNAME, NO_OF_ORDERS, and AVG_ORDER_AMT, where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer. iv) List the Order# for the orders that were shipped from all the warehouses that the company has in a specific city. v) Demonstrate how you delete Item# 10 from the ITEM table and make that field <i>null</i> in the ORDER-ITEM table. vi) Generation of suitable reports. vii) Create a suitable front end for querying and displaying the results.</p>
26.	<p>Consider the following database of student enrollement in courses and books adopted for each course.</p> <p>STUDENT (regno: string, name: string, major: string, bdate: int) COURSE (course#: int, cname: string, dept: string) ENROLL (regno : string , course#: int , sem : int , marks : int) BOOK_ADOPTION (course#: int, sem: int, book_isbn: int) TEXT (book_isbn: int, book-title: string, publisher: string, author: string).</p> <p>i) Create the above tables by properly specifying the primary keys and the foreign key. ii) Enter at least five tuples for each relation. iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department. iv) Produce a list of text books (include course #, book_isbn,book-title) in the alphabetical order for courses offered by the cs department that use more than 2 books. v) List any department that has all its adopted books published by specific publisher. vi) Generation of suitable reports.</p>
27.	<p>Consider the following relations for the details maintained by a book dealer.</p> <p>AUTHOR (Author-id: int, Name: string, City: string, Country: string) PUBLISHER (Publisher-id: int, Name: string, City: string, Country: string) CATALOG (Book-id: int, title: string, author-id: int, Publisher-id: int, Category-id: int, Year: int, Price: int) CATEGORY (Category-id: int, Description: string) ORDER-DETAILS (Order-no:int, Book-id: int, Quantity: int)</p> <p>i. Create the above tables by properly specifying the primary keys and the foreign keys. ii. Enter at least five tuples for each relation. iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000. iv. Find the author of the book which has maximum sales. v. Demonstrate how you increase the price of books published by a specific publisher by 10%. vi. Generation of suitable reports. vii. Create a suitable front end for querying and displaying the results.</p>

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Discrete Mathematics
Semester: 1st
Credits: 04
Pre Requisite: -----

Course Code: MIT503C
Paper Type: Core
Max Marks: 100
Co-Requisite: MIT501C

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To familiarize students about set theory concepts, operations and uses in computer sciences.*
- *To explain concept of Discrete Structures which include Semigroups, Monoids, and Groups together with their uses in computer science and their corresponding structures in computer science*
- *To explain students concepts like Partially Ordered Sets, Lattices and Finite Boolean Algebra and use in Computer Science.*
- *Give student a knowledge about finite state Machines and possible modeling of process using finite state Machine concepts.*

COURSE CONTENT:

UNIT I

Proposition, Logic, Truth tables, Propositional Equivalence, Logical Equivalence, Predicates & Quantifiers, Sets: operations on sets, Computer representation of sets, Functions: Domain, Range, One-to-One, Onto, Inverses & Composition, Cardinality of a Set, sequences & summations, The growth of functions . Methods of Proof: Different methods of proof, Direct Proof, Indirect Proof, Mathematical Induction for proving algorithms.

UNIT II

Discrete probability, Advanced Counting Techniques: Inclusion-Exclusion, Applications of inclusion-exclusion principle, recurrence relations, solving recurrence relation. Relations: Relations & their properties, Binary Relations, Equivalence relations, Diagraphs, Matrix representation of relations & digraphs, Computer representation of relations & digraphs, Transitive Closures, Warshall's Algorithm.

UNIT III

Partially Ordered Sets (Posets), External elements of partially ordered sets, Hasse diagram of partially ordered set, isomorphic ordered set, Lattices: Properties of Lattices, complemented Lattices. Graph theory: Introduction to graphs, Graph Terminology Weighted graphs, Representing Graphs, Connectivity of Graphs: Paths & Circuits, Eulerian & Hamiltonian Paths, Matrix representation of graphs. Graph Coloring. Cyclometric Complexity.

UNIT IV

Trees: Rooted trees, Application of trees: Binary Search Trees, Decision Trees, Prefix Codes, Tree traversal, trees & sorting, spanning trees, minimal spanning trees. "Catalans Series (Number), B+ Trees, Red-Black Trees Finite Boolean algebra, Functions on Boolean algebra, Boolean functions as Boolean polynomials. Groups & applications: Subgroups, Semigroups, Monoids, Product & quotients of algebraic structures, Isomorphism, Homomorphism

COURSE OUTCOMES:

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

- *Develop mathematical and logical thinking*
- *Express real life problems it in terms of predicates, quantifiers, and logical connectives and obtained its solution*
- *Utilize the concepts of relations and functions to solve simple real life problems.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. KENNETH H. ROSEN “Discrete Mathematics & Its Applications, Tata McGraw – Hill
2. Ralph P. Grimaldi, Discrete And Combinatorial Mathematics An Applied Introduction

References:

1. LIU “Elements of Discrete Mathematics “ Tata McGraw Hill
2. SCHAUMS “Discrete Mathematics“ Tata McGraw Hill
3. KOLMAN/REHMAN “Discrete Mathematical Structures“ Pearson Education
4. NICODEMI “Discrete Mathematics“ CB



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Computer Organization and Architecture	Course Code:	MIT504C
Semester:	1 st	Paper Type:	Core
Credits:	04	Max Marks:	100
Pre Requisite:	-----	Co-Requisite:	MIT501C, MIT511E
Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To know and understand the main components of a computer system and the considerations in their design.*
- *To acquire tools for comparison among alternatives.*
- *To know and understand performance measures, as well as their impact on system architecture.*
- *To understand the interplay among system components, design trade-offs, etc.*

COURSE CONTENT:

UNIT I

Fundamental concepts of computer architecture & organization, Register Organization. Interconnection Structures, Bus Interconnections, Integer/Floating Point Arithmetic & Representation, Instruction Cycle & Interrupts. Instruction Set Characteristics & Functions. Addressing Modes & Formats.

UNIT II

Processor Organization: ALU, Design of Arithmetic Circuit, Design of Logic Circuit & Design of ALU. Processor Unit: Design of Accumulator. Control Organization: Hardwired / Micro-Programmed Control, Control Memory, Address Sequencing, Design of Control Unit & Micro-Program Examples.

UNIT III

Memory Hierarchy, Main Memory: RAM/ROM Chips. Memory Address Map, Memory Connection to CPU, Associative Memory, Hardware Organization – Match Logic, Read/Write Operation, Cache Memory, Virtual Memory, Memory Management – Associated Hardware, I/O Organization. Peripheral Devices: I/O Interfaces, Asynchronous Data Transfer, Modes of Transfer, Direct Memory Access & I/O Processor.

UNIT IV

RISC Architectures – their characteristics & comparison to CISC, Introduction to Parallel Processing, Basic Parallelization Techniques. Pipelining – Arithmetic & Instruction Pipelining, RISC Pipeline, Vector & Array Processors. Multiprocessor System Architectures & their Characteristics, Interconnection Structures, Inter-processor Arbitration, Inter-processor Communication & Synchronization, Cache Coherence Issues. Introduction to Super Scalar Processors

COURSE OUTCOMES:

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

- *Get familiarized with basics of computer hardware and how software interacts with computer hardware.*
- *Understand how computers represent and manipulate data, computer arithmetic and conversion between different number systems.*
- *Understand how Boolean algebra is related to designing computer logic, through simple combinational and sequential logic circuits.*
- *Understand basics of Instruction Set Architecture (ISA).*
- *Understand with a simple computer with hardware design including data format, instruction format, instruction set, addressing modes, bus structure, input/output, memory, Arithmetic/Logic unit, control unit, and data, instruction and address flow.*
- *Design combinational and sequential logic circuits, flip-flops, counters, shift registers, adders, subtractor, multiplexer, demultiplexer, Arithmetic/Logic unit.*
- *Understand concept of memory unit and input/output architecture.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. W. Stallings, “computer organization & architecture”.

References:

1. V.C. Hamacher, A.G. Vranesic and S.G. Zaky, “Computer Organization”, Tata McGraw Hill
2. J.P Hayes, “Computer Architecture & Organization”, Tata McGraw Hill
3. Morris Mano, “Computer System Architecture”, PHI.
4. M. J. Flynn, “Computer Architecture”, Narosa
5. David A. Patterson, John L. Hennessey, “ Computer Organization”
6. GovindaRajalu, “Computer Architecture & Design ” TMH



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Digital Electronics
Semester: 1st
Credits: 03
Pre Requisite: -----

Course Code: MIT511E
Paper Type: DCE
Max Marks: 75
Co-Requisite: MIT504C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVE:

- *To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.*
- *To prepare students to perform the analysis and design of various digital electronic circuits.*
- *The course also covers digital logic design intended to make students familiar with different types of designs as sequential logic circuits and combinational logic circuits.*

COURSE CONTENT:

UNIT I

Introduction to Digital Systems, Number systems: Binary number system, Octal & Hexa-decimal number system, Conversion of Number System, Arithmetic operations on Binary numbers, Number representation: Signed Binary numbers, 1's and 2's Complement. Fixed- and Floating-Point Representation. Logic Gates: AND, OR, NOT NOR, NAND & XOR & their Truth tables Logic diagrams. Boolean Postulates and Laws- Principal of Duality.

UNIT II

Minimization of Boolean Expression: Introduction, Minterm, Maxterm, Sum of Products (SOP)- Product of Sums (POS), Karnaugh map Minimization upto 4 variables. Don't Care Conditions. Error Detection and Correction: single parity and block parity check, hamming code, Combinational Logic: Introduction to Adders, Half Adder, Full Adder, Parallel binary adders, Subtractors- Half Subtractor, Full Subtractor, Multiplexer, Demultiplexer, Decoder, Encoder.

UNIT III

Sequential Logic - Definition & Basic sequential circuits, Latches & Flip-flops: SR-Flip Flop, D-Flip Flop, JK flip-flop, T flip-flop, Race condition, JK Master Slave Flipflop, Timing Specifications, Counters: Basics of Asynchronous & Synchronous counters, Registers, Shift Registers, Types of Shift registers.

COURSE OUTCOMES:

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

- *To understand and examine the structure of various number systems and its application in digital design.*
- *Understand, analyze and design various combinational and sequential circuits.*
- *Identify basic requirements for a design application and propose a cost effective solution.*
- *Identify and prevent various hazards and timing problems in a digital design.*
- *To develop skill to build, and troubleshoot digital circuits*

Text Books:

1. Digital Logic and Computer Design by M Morris Mano

References:

1. Digital Fundamentals, Global Edition (Kindle Edition) by Thomas L Floyd.
2. Modern Digital Electronics, 3rd Edition by R.P Jain
3. Digital Electronics: principles, Devices & Applications, Wiley by Anil K Maini.

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Green Technologies

Semester: 1st

Credits: 03

Pre Requisite: -----

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT512E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT504C, MIT526E

COURSE OBJECTIVES:

- *Introduce students to the fundamental concepts of multimedia, including text, images, audio, video, and animation, as well as their integration into cohesive multimedia presentations.*
- *Familiarize students with various multimedia elements such as images (e.g., raster vs. vector graphics), audio (e.g., digital audio formats, compression techniques), video (e.g., codecs, containers), and interactive elements.*
- *Equip students with the skills needed to create, edit, and manipulate digital multimedia content using appropriate software tools and techniques. This may include image editing software (e.g., Adobe Photoshop), audio editing software (e.g., Audacity), video editing software (e.g., Adobe Premiere Pro), and multimedia authoring tools (e.g., Adobe Animate).*
- *Teach students how to integrate different multimedia elements into cohesive presentations or applications and add interactivity using scripting languages or multimedia authoring environments.*
- *Explain the principles of multimedia compression and optimization techniques to reduce file sizes without sacrificing quality. Discuss the importance of optimizing multimedia content for efficient delivery over different networks and devices.*

COURSE CONTENT:

UNIT I

Introduction to Multimedia. Definition, Components of multimedia, Multimedia Past and Present; History, Hypermedia, WWW and Internet, Multimedia Today. Users of multimedia, Applications of Multimedia; Education and Training, Entertainment and Art Industry, Engineering, Research, Communication, Medicine, Marketing and Advertising, Journalism, Sports, Gaming.

Graphics and Images: Image Properties; Resolution, Bit Depth, Color, Aspect Ratio, Dpi, Pixels. Popular Image File Formats: JPEG, GIF, PNG, TIFF. Basics of Image Capture.

UNIT II

A Multimedia System. Features of a Multimedia System, Multimedia PC workstation components; Input Devices / Capture devices, Storage Devices, Communication Ports, Output / Display Devices. Graphics Processor (Internal and external). Examples of Various Multimedia Systems.

UNIT III

Multimedia Configuration: Converging technologies, Functions & subsystems (input, development & output). Multimedia Development Tools: Presentation Tools, Authoring Tools and relevant Examples. Data Compression.

Multimedia Video: Frame Rate, Display; interlaced vs Non-interlaced. Video File Formats: MPEG, AVI.

Multimedia Audio: Digital and Analog, Some Audio File Formats. Animation Basics.

COURSE OUTCOMES:

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

- *Understanding Multimedia Concepts: Students will be able to define multimedia and explain its basic components, including text, images, audio, video, and animation.*
- *Media Creation and Editing Skills: Students will demonstrate proficiency in creating, editing, and manipulating multimedia content using appropriate software tools, such as image editing software, audio editing software, and video editing software.*
- *Integration and Interactivity: Students will be able to integrate multiple media types into cohesive multimedia presentations or applications and add interactivity using scripting languages or multimedia authoring environments.*
- *Students will understand multimedia compression techniques and be able to optimize multimedia content for efficient delivery over different networks and devices.*

MSc. IT Syllabus – Department of Computer Science, IUST

- *Students will demonstrate awareness of legal and ethical issues related to multimedia content creation, distribution, and consumption, including copyright law, fair use, privacy concerns, and digital rights management.*

Text Books:

1. Fundamentals of Multimedia Ze-Nian Li, Mark S.Drew, Jiangchuan Liu, 2nd Edition (Pearson)
2. Multimedia in Practice Judith Jeffcote (PHI).

References:

1. Buford, John F.K, "Multimedia Systems", Pearson Education.
2. Kerman, P., "Sams Teach Yourself Macromedia Flash 8 in 24 Hours", 3rd Edition, Pearson Education.
3. Reinhardt, R., Dowd, S., "Macromedia Flash 8 Bible 1st Edition" , 1st Edition, Wiley.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Management Information System

Semester: 1st

Credits: 03

Pre Requisite: -----

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT513E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT502C

COURSE OBJECTIVES:

- *The objective of Course Management Information System is to bring a systematic Knowledge of Management Information technology so that it can be appreciated and understood for application in business and industry, thus more emphasis is laid to application of information to business management and thus provides communication with employees, maintain employee records and coordinate work activities.*

COURSE CONTENT:

UNIT I

Overview of Management Information Systems: Introduction, Concepts & characteristics, Components of MIS, Role of MIS, Concepts of Data, Information, Knowledge & Intelligence;

Framework for Understanding MIS: Robert Anthony's hierarchy of management, MIS: Support to management, Relatedness of MIS with management activities, Attributes of Information system, Relationship between Organization & Information systems.

UNIT II

Introduction to decision making: Structured versus unstructured Decisions, Managerial Decision Making Process, Types of Decisions, Simons Model of decision making, Decision Support Systems: Overview, Concepts, Characteristics & Components, Overview of MBMS

UNIT III

Expert system: Basic Concepts, Comparison of conventional & expert systems, Structure of expert system, "Comparison of ES and DSS." Executive support system: Needs & characteristics, Role of executive support system in organization.

COURSE OUTCOMES:

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

- *Understand the scope & importance of Management Information Systems*
- *Understand the concept and applications of DBMS, Systems Engineering Analysis and Design*
- *Understand & Apply Decision Support Systems Models with Digital Firm*

Text Books:

1. Laudon, "Management Information Systems", Pearson
2. Jawadekar, "Management Information Systems",

References:

1. Kroenke, Management Information Systems", Mc-Graw Hill.
2. Mudrick R.G., Ross, J.E. and Gleggt, J.R."Information Systems for Modern Management", PHI.
3. Jayashankar: Decision Support Systems, Mc-Graw Hill.
4. Stephen Haag & Maeve Cummings, "Information Systems Essentials", McGraw-Hill
5. Ralph Stair, "Principles of Information Systems", ISBN: 0619064897

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Web Technologies
Semester: 1st
Credits: 03
Pre Requisite: -----

Course Code: MIT526E
Paper Type: DCE
Max Marks: 75
Co-Requisite: MIT501C, MIT512E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To provide knowledge on web architecture, web services, client side and server side scripting technologies*
- *To focus on the development of web-based information systems and web services.*
- *To design interactive web pages using HTML and Style sheets.*
- *To provide skills to design interactive and dynamic web sites.*
- *To provide knowledge for implementing web applications using JavaScript.*

COURSE CONTENT:

UNIT I

Web Essentials: Clients, Servers, and Communication. The Internet-Basic Internet Protocols, Introduction, Concept of Internet- History of Internet, Protocols of Internet, World Wide Web, Web browsers, web servers, MIME, URL, HTTP , URL, Web Clients ,Web Server , Web sites, Web Applications, types of Web applications, HTTP request-response message.

UNIT II

HTML- Introduction, History of HTML, Versions-Basic XHTML Syntax and Semantics, Structure of HTML Document: HTML Tags, open and closed tags, HTML Editors, Elements, Attributes, Styles, Formatting, working with formatting, form, tables, frames, links, images, lists, multimedia: Audio, video, Graphics: Canvas and SVG, HTML APIs: Geolocation, Drag/Drop, web storage.

UNIT III

Cascading Style Sheets: Introduction, syntax, selectors, colors, backgrounds, borders, Margins, padding, outline, text, Fonts, Icons, Lists, Tables, Display, Max-width, position, CSS Responsive: Viewport, Grid-view, media Queries, Images, video, frameworks and templates. Introduction to Javascript, controls statements, Arrays and functions, pattern matching, Element Access, Event Handling

COURSE OUTCOMES:

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

- *Understand & Apply (X)HTML(5)+CSS programming*
- *Demonstrate dynamic webpage development using java script and DHTML*
- *Design a well formed / valid XML document*
- *Create a server side ASP application using database*

Text Books:

1. HTML5 Black Book by Dreamtech
2. HTML & CSS: Design and Build Web Sites by Jon Duckett, Wiley
3. Web Programming By Chris Bates , Wiley Publications

References:

1. Mastering HTML, CSS & Javascript Web Publishing by Laura Lemay, BPB Publications
2. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, by Dreamtech Press
3. HTML & CSS: The Complete Reference, Fifth Edition by Thomas Powell, McGraw Hill Education

MSc. IT Syllabus – Department of Computer Science, IUST

List of Practicals:

LAB JOURNAL_MIT526E	
S. No.	TITLE
1.	Create a simple webpage using HTML.
2.	Basic HTML Tags, Table Tags, List Tags, Image Tags, Forms.
3.	Use new HTML5 form elements Email, Password, Files, Date, Number, Data list
4.	Include Audio, Videos in HTML pages
5.	Experimenting with CANVAS
6.	Drag and Drop in Webpages
7.	Incorporating Messaging feature in Websites;
8.	Add a Cascading Style sheet for designing the web page.
9.	Adding inline, internal and external CSS
10.	CSS Transitions
11.	CSS Animation
12.	Margin, Padding and Borders in CSS
13.	Background images in CSS
14.	Design a dynamic web page with validation using JavaScript.
15.	Use user defined function to get array of values and sort them in ascending order
16.	Demonstrate String and Math Object's predefined methods
17.	Demonstrate Array Objects and Date Object's predefined methods
18.	Exception Handling
19.	Calendar Creation : Display all month
20.	Write an HTML page that contains a selection box with a list of 5 countries In the above page when the user selects a country, its capital should be printed next to the list, and add CSS to customize the properties of the font of the capital.
21.	A web application that takes name and age from an HTML page. If the age is less than 18, it should send a page with "hello <name>,and u are not authorized to visit this site" otherwise it should send "welcome <name> to this site" message where name should be replaced with the entered name otherwise it should send welcome<name> to the site message
22.	WAP in JS to accept data from the user and display the same
23.	WAP in JS to demonstrate various I/O functions.
24.	WAP in JS to demonstrate the type conversion
25.	WAP in JS to perform various string manipulations
26.	WAP in JS to demonstrate the use of Objects, Arrays, Maps, Sets
27.	WAP in JS to create an animation
28.	WAP in JS to test if the number is even or odd
29.	WAP in JS to demonstrate the usage of conditional constructs
30.	WAP in JS to demonstrate the implementation of looping constructs
31.	WAP in JS to demonstrate the implementation of bitwise operators
32.	WAP in JS to demonstrate the implementation of logical operators
33.	WAP in JS to find the factorial of a number
34.	WAP in JS to check if a number is prime or composite

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: E-Commerce
Semester: 1st
Credits: 03
Pre Requisite: -----

Course Code: MIT527E
Paper Type: DCE
Max Marks: 75
Co-Requisite: MIT526E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To provide basic concepts of e-business and e-commerce, including presentation and discussion of the strategies and technologies involved.*
- *To provide theoretical and practical issues of conducting business over the internet and the Web*
- *To present methods for evaluating user needs.*
- *To provide an understanding of E-business Infrastructure, Selling and Marketing on the Web, Web Server Hardware and Software, Business-to-Business strategies, Virtual Communities, Web Portals, E-commerce Software, Payment systems, Security and User Experience.*

COURSE CONTENT:

UNIT I

E-Commerce (Introduction and Definition), Goals of E-Commerce, Technical Components, Functions, Advantages and Disadvantages, Applications.

The Internet and WWW - Evolution of Internet, Domain Names and Internet Organisation (.edu, .com, mil-gov, .net etc), Internet Service provider.

UNIT II

E commerce business models: key elements of business model, Business to Consumer (B2C) model, B2B model, consumer to consumer model(C2C).Building E commerce- system development life Cycle, choosing software and hardware e commerce, site tools. Benefits of website, Registering a Domain Name, Web promotion. Internet Security, Secure Transaction, Computer Crime (types of Crimes), Threats.

UNIT III

Electronic Data Interchange, Introduction, Concepts of EDI and Limitation, Application of EDI, Disadvantages of EDI, EDI model, Electronic payment System, Introduction, Types, Strategies for developing electronic commerce web sites, Net marketplaces-characteristics of net marketplaces, types of net marketplaces, E distributors, E procurement, Exchanges. Online content providers-digital copyrights and electronic publishing.

COURSE OUTCOMES:

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

- *Understand the basic business management concepts, technical concepts, legal issues, and privacy relating to E-commerce.*
- *Understand how E-commerce is affecting business enterprises, governments, consumers and people in general.*
- *Describe the infrastructure needed for E-commerce and various electronic payment systems.*

Text Books:

1. E-Commerce Concepts, Models, Strategies by G.S.V Murthy
2. E commerce: by Laudon

References:

1. E-Commerce by Kamlesh K Bajaj and Debjani Nag
2. Electronic Commerce by Gary P. Schneider

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Multimedia Technologies

Semester: 1st

Credits: 03

Pre Requisite: -----

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT528E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT504C, MIT541A

COURSE OBJECTIVES:

- *Introduce students to the fundamental concepts of multimedia, including text, images, audio, video, and animation, as well as their integration into cohesive multimedia presentations.*
- *Familiarize students with various multimedia elements such as images (e.g., raster vs. vector graphics), audio (e.g., digital audio formats, compression techniques), video (e.g., codecs, containers), and interactive elements.*
- *Equip students with the skills needed to create, edit, and manipulate digital multimedia content using appropriate software tools and techniques. This may include image editing software (e.g., Adobe Photoshop), audio editing software (e.g., Audacity), video editing software (e.g., Adobe Premiere Pro), and multimedia authoring tools (e.g., Adobe Animate).*
- *Teach students how to integrate different multimedia elements into cohesive presentations or applications and add interactivity using scripting languages or multimedia authoring environments.*
- *Explain the principles of multimedia compression and optimization techniques to reduce file sizes without sacrificing quality. Discuss the importance of optimizing multimedia content for efficient delivery over different networks and devices.*

COURSE CONTENT:

UNIT I

Introduction to Multimedia. Definition, Components of multimedia, Multimedia Past and Present; History, Hypermedia, WWW and Internet, Multimedia Today. Users of multimedia, Applications of Multimedia; Education and Training, Entertainment and Art Industry, Engineering, Research, Communication, Medicine, Marketing and Advertising, Journalism, Sports, Gaming. Graphics and Images: Image Properties; Resolution, Bit Depth, Color, Aspect Ratio, Dpi, Pixels. Popular Image File Formats: JPEG, GIF, PNG, TIFF. Basics of Image Capture.

UNIT II

A Multimedia System. Features of a Multimedia System, Multimedia PC workstation components; Input Devices / Capture devices, Storage Devices, Communication Ports, Output / Display Devices. Graphics Processor (Internal and external). Examples of Various Multimedia Systems.

UNIT III

Multimedia Configuration: Converging technologies, Functions & subsystems (input, development & output). Multimedia Development Tools: Presentation Tools, Authoring Tools and relevant Examples. Data Compression. Multimedia Video: Frame Rate, Display; interlaced vs Non-interlaced. Video File Formats: MPEG, AVI. Multimedia Audio: Digital and Analog, Some Audio File Formats. Animation Basics.

COURSE OUTCOMES:

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

- *Understanding Multimedia Concepts: Students will be able to define multimedia and explain its basic components, including text, images, audio, video, and animation.*
- *Media Creation and Editing Skills: Students will demonstrate proficiency in creating, editing, and manipulating multimedia content using appropriate software tools, such as image editing software, audio editing software, and video editing software.*
- *Integration and Interactivity: Students will be able to integrate multiple media types into cohesive multimedia presentations or applications and add interactivity using scripting languages or multimedia authoring environments.*
- *Students will understand multimedia compression techniques and be able to optimize multimedia content for efficient delivery over different networks and devices.*
- *Students will demonstrate awareness of legal and ethical issues related to multimedia content creation, distribution, and consumption, including copyright law, fair use, privacy concerns, and digital rights management.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

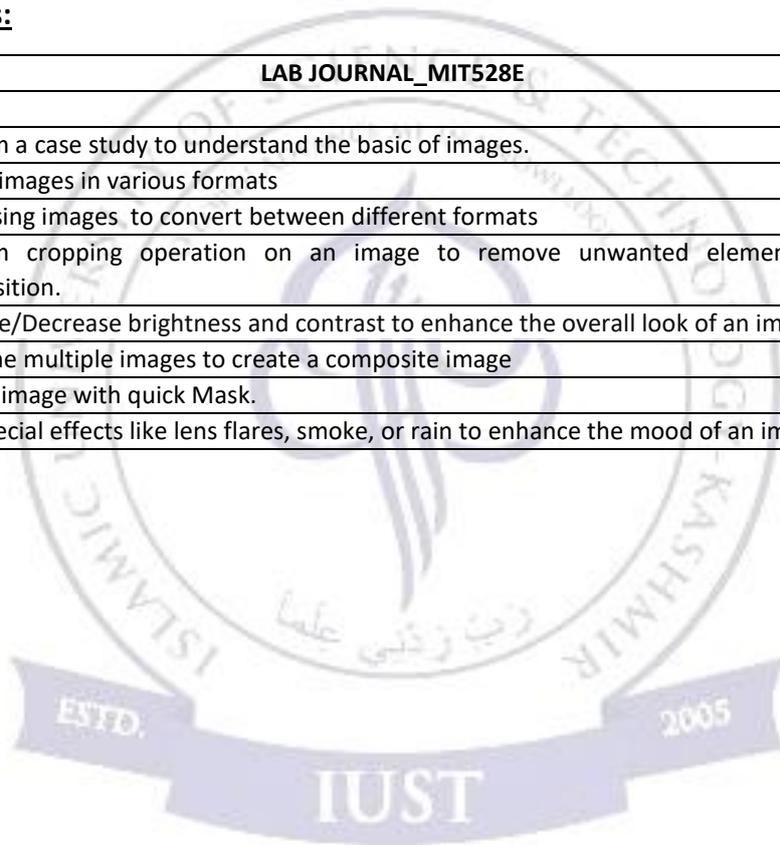
1. Fundamentals of Multimedia Ze-Nian Li, Mark S.Drew, Jiangchuan Liu, 2nd Edition (Pearson)
2. Multimedia in Practice Judith Jeffcote (PHI).

References:

4. Buford, John F.K, "Multimedia Systems", Pearson Education.
5. Kerman, P., "Sams Teach Yourself Macromedia Flash 8 in 24 Hours", 3rd Edition, Pearson Education.
6. Reinhardt, R., Dowd, S., "Macromedia Flash 8 Bible 1st Edition" , 1st Edition, Wiley.

List of Practicals:

LAB JOURNAL_MIT528E	
S. No.	TITLE
1.	Perform a case study to understand the basic of images.
2.	Create images in various formats
3.	Processing images to convert between different formats
4.	Perform cropping operation on an image to remove unwanted elements or to improve composition.
5.	Increase/Decrease brightness and contrast to enhance the overall look of an image.
6.	Combine multiple images to create a composite image
7.	Editing image with quick Mask.
8.	Add special effects like lens flares, smoke, or rain to enhance the mood of an image.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Information and Communications Technology	Course Code:	MIT541A
‘Semester:	1 st	Paper Type:	BC
Credits:	00	Max Marks::	50
Pre Requisite:	-----	Co-Requisite:	MIT526E, MIT512E
Marks Distribution: (Mid Term:15, End Term:25, Viva:05, Assignment / Presentations:05)			

COURSE OBJECTIVE:

- *The course introduces the students to basic computer concepts. Emphasis of the course is on providing the students with an introduction to programming, programming paradigms, database, basic networking and security. The course also covers digital logic design intended to make students familiar with different types of designs as sequential logic circuits and combinational logic circuits. Widely used software’s including word processing, spreadsheets and presentations are studied. Main objectives of the course are to build an appreciation for the fundamental concepts of computers and to become familiar with popular PC productivity software.*

COURSE CONTENT:

UNIT I

Introduction to Computer: Generations and Characteristics.

Components: Hardware – Input devices (OMR, OCR, MICR, etc.). Output devices, Central Processing Unit, Memory- Primary and Secondary. Software - Introduction, Types – System and Application.

Operating system: Definition, Functions, Types, Classification, Elements of command based and GUI based operating system.

Computer Network: Definition, criteria, process of communication, Types (LAN, WAN and MAN), Network topologies, devices. Transmission media: guided and unguided media.

UNIT II

Problem solving concept: Algorithms – Definition, Characteristics, Limitations.

Computer Arithmetic & Number System: Positional number system – Binary, Octal and hexadecimal number systems; Methods of base conversions; Computer Arithmetic.

Database: Overview, features, design, Database models –RDBMS.

Internet: Overview, Architecture, Functioning, Basic services like WWW, E-mail, Web Browsers.

Overview of Emerging Technologies: Artificial intelligence, Big-data analytics, Cloud computing, Internet of Things (IoT), Block chain technology.

Role of IT in different areas – Education, industry, banking, marketing, healthcare.

COURSE OUTCOMES:

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

- *Understand the basic concepts of computer and its component*
- *Relate theory and practice of computer architecture*
- *Illustrate the basic data representation in the computer*
- *Comprehend the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit*

Text Books:

1. Rajaraman V., “Fundamentals of Computers”, Prentice-Hall of India.
2. Norton P., “Introduction to Computers”, McGraw Hill Education.
3. Goel A., “Computer Fundamentals”, Pearson.

MSc. IT Syllabus – Department of Computer Science, IUST

4. Balagurusamy E., “Fundamentals of Computers”, McGraw Hill.
5. Thareja R., “Fundamentals of Computers”, Oxford University Press.
6. Bindra J., “The Tech Whisperer- on Digital Transformation and the Technologies that Enable it”, Penguin.

References:

1. V. Rajaraman, “Introduction to Computers”, TMH
2. Peter Norton, “Introduction to Computers”, TMH
3. Suresh K. Basandra, “Computers Today, 2005”, Galgotia Publications.





Semester - II

MSc. IT Syllabus – Department of Computer Science, IUST

Semester-II (26 Credit Semester with 33 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT551C	Data Structures	Core	5	3	0	4	07	FC
MIT552C	Introduction to Python	Core	5	3	0	4	07	SEC
MIT553C	Data Communication & Computer Networks	Core	4	3	1	0	04	FC
MIT554C	Operating System	Core	4	3	1	0	04	FC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MIT561E	Web Mining	DCE	3	2	1	0	03	SEC
MIT562E	Quantitative Techniques	DCE	3	2	1	0	03	AEC
MIT563E	Microprocessor & Assembly Language Programming	DCE	3	2	1	0	03	AEC
MIT564E	Internet of Things	DCE	3	2	1	0	03	SEC
Pool B: 03 Discipline Centric Elective Credit Units								
MIT576E	Open Source Technologies	DCE	3	2	0	2	04	EOC
MIT577E	Computer Graphics	DCE	3	2	0	2	04	SEC
MIT578E	Numerical and Statistical Computing	DCE	3	2	0	2	04	VAC
MIT579E	Pervasive Computing	DCE	3	2	0	2	04	AEC
Audit Courses (Mandatory)								
MIT591A	Technical Communication	Audit	0	2	0	0	02	VAC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			26				33	

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Data Structures

Semester: 2nd

Credits: 05

Pre Requisite: MIT501C, MIT503C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

Course Code: MIT551C

Paper Type: Core

Max Marks:: 125

Co-Requisite: MIT552C

COURSE OBJECTIVE:

- *The objective of this course is to introduce the students to the significant topics of basic as well as advanced data structures and subsequently make them understand & learn the commonly used data structures along with their applications. The focus is also to appreciate the need and working of different ways of storing data and demonstrate the advantages and disadvantages of specific data structures both linear as well as non-linear.*
- *To emphasizes on lab work wherein the students learn not only to make use of different data structures, but also their application in different synthetic problems. The practical demonstration for autonomous realization of simple programs or program parts is to understand the behavior of these basic data structures and to have a thorough understanding of how data structures influence the performance of algorithms.*

COURSE CONTENT:

UNIT I

Data Structures: Overview & Significance. Introduction & Analysis of Algorithms. Linear & Non Linear Data Structures. Array & Strings, Application of Arrays, Sparse Matrix, Searching & Sorting: Linear & Binary Search, Bubble Sort, Selection Sort & Insertion Sort. Recursion & Applications.

UNIT II

Stack: Operations & Applications, Polish Expressions Queue: Operations & Applications, Circular Queue: Operations & Applications. Linked List: Singly Linked List, Doubly Linked List, Circular Linked List. Linked implementation of Stack & Queue, Operations & Applications of Linked Lists.

UNIT III

Trees: Basic Concept, Terminology & Representation of a Tree. Binary Tree, Binary Search Tree, Tree Traversal Techniques, BST Sort, AVL Trees, Heap Sort, Quick Sort, Merge Sort. Threaded Binary Tree, Red Black Trees, B Trees, B* Trees, Applications of Trees.

UNIT IV

Graph: Matrix & List Representation, Elementary Graph Operations (BFS & DFS), Single Source Shortest Path, Dijkstras Algorithm, Bellman-Ford Algorithm. All Pair Shortest Paths, Transitive

Closer, Floyd-Warshall Algorithm. Spanning Trees: Minimum Cost Spanning Trees, Prims & Kruskal Algorithm. Hashing: Overview, Hashing Functions, Collision Resolution Techniques.

COURSE OUTCOMES:

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

- *Develop some simple applications, like a desk calculator using stacks.*
- *Understand advanced searching methods like B-tree, B+ tree, AVL/red-black trees.*
- *Learn to develop a basic file system.*
- *Use standard libraries for data structures.*

Text Books:

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, 2nd Ed.

References:

School of Engineering & Technology (SoE&T),

Islamic University of Science & Technology (IUST),

1-University Avenue, Awantipora, Pulwama, Pin -192122, Jammu and Kashmir.

Page - 36

MSc. IT Syllabus – Department of Computer Science, IUST

1. Data structures and Algorithms in C++ -- by Adam Drozdek (1994 2001).
2. Fundamentals of Data Structures in C -- by Horowitz, Sahni and Anderson-Freed (Silicon Press 2007).
3. Data Structure Using C and C++ -- by Y. Langsam, M. J. Augenstein and A. N. Tanenbaum (Pearson Education, 2nd Edition, 2015).

List of Practicals:

LAB JOURNAL_MIT551C	
S. No.	TITLE
1.	Create One dimensional array of Integers and then perform the following operations on it: <ol style="list-style-type: none"> i. Read the Elements in an Array ii. Display the Elements in Input Order iii. Display the Elements in Reverse Order iv. Search a Particular Element v. Replace a Particular Element vi. Insert a Missed Element vii. Sort the Array Elements in any Order viii. Insert a Missed Element (after sorting) ix. Delete a Particular Element x. Frequency of a Particular Element xi. Display the Distinct Elements xii. Frequency of Distinct Elements xiii. Display the Duplicate Elements xiv. Remove the Duplicate Elements xv. Display the sum of all Elements xvi. Get the Average of all Elements xvii. Display the Maximum Value Element xviii. Display the Minimum Value Element
2.	Create a Two dimensional array of Integers and then perform the following operations on it: <ol style="list-style-type: none"> i. Read the elements of the 2D array ii. Display the elements iii. Get Transpose of Matrix (2D Array) iv. Perform the addition & subtraction of two matrices(2D arrays) v. Perform the Multiplication of Two Matrices(2D arrays)
3.	Generate the information matrix from a Sparse Matrix and then regenerate the Original Matrix.
4.	Perform a Linear Search as well as the Binary Search on an array of Integers
5.	Perform Bubble Sort & Insertion Sort on an array of Integers
6.	Create a Stack and perform all the permissible operations of Stack
7.	Create a Simple Queue and perform the permissible operations of Simple Queue
8.	Create a Circular Queue and perform the permissible operations of Circular Queue
9.	Convert an Infix expression to its corresponding postfix expression and then evaluate it as an application of Stack.
10. K	Get the Factorial of a number using Recursion.
11.	Create a Singly Linked List and perform the following operations on it: <ol style="list-style-type: none"> i. Display the Node Information in Input Order ii. Search a Particular Node iii. Replace a Particular Node Information iv. Insert a Missed Node Information (at the Beginning, at Specific Location or at the end) v. Sort the Node Information in any Order(Prefer Insertion Sort)

MSc. IT Syllabus – Department of Computer Science, IUST

	<ul style="list-style-type: none"> vi. Sort the Nodes based on Address exchange vii. Insert a Missed Node(after sorting) viii. Delete a Particular Node (at the Beginning, at Specific Location or at the end) ix. Find Frequency of a Particular Node Information x. Reverse the List
12.	<p>Create a Doubly Linked List and perform the following operations on it:</p> <ul style="list-style-type: none"> i. Display the Node Information in Input Order ii. Search a Particular Node iii. Replace a Particular Node Information iv. Insert a Missed Node Information (at the Beginning, at Specific Location or at the end) v. Sort the Node Information in any Order(Prefer Insertion Sort) vi. Sort the Nodes based on Address exchange vii. Insert a Missed Node (after sorting) viii. Delete a Particular Node (at the Beginning, at Specific Location or at the end) ix. Find Frequency of a Particular Node Information x. Display the Node Information in Reverse Input Order
13.	<p>Create a Circular Linked List and perform the following operations on it:</p> <ul style="list-style-type: none"> i. Display the Node Information ii. Search a Particular Node iii. Replace a Particular Node Information iv. Insert a Missed Node Information v. Sort the Node Information in any Order vi. Sort the Nodes based on Address exchange vii. Insert a Missed Node (after sorting) viii. Delete a Particular Node ix. Find Frequency of a Particular Node Information x. Display the Node Information in Reverse Order
14.	Create a Linked Stack and perform all the permissible operations of Stack
15.	Create a Linked Queue and perform all the permissible operations of Queue
16.	Create a Binary Tree and use the traversal Techniques to display the nodes/vertices of Binary Tree?
17.	<p>Create a Binary Search Tree (BST) and then perform the following operations on it:</p> <ul style="list-style-type: none"> i. Visit the nodes of BST (InOrder, PreOrder&PostOrder) ii. Delete any particular leaf node iii. Delete a node with only One SubTree (Left or Right) iv. Delete a node with both SubTrees
18.	Perform Selection Sort, Quick Sort & Merge Sort on an array of Integers.
19.	<p>Perform the following operations on an AVL Tree:</p> <ul style="list-style-type: none"> i. Insert an element into a AVL tree. ii. Delete an element from a AVL tree. iii. Search for a key element in a AVL tree.
20.	Perform Binary Search Sort & Heap Sort on a set of integers in a heap.
21.	Demonstrate the Threaded Binary Tree and perform all the permissible operations on it.
22.	Demonstrate the Red Black Tree and perform all the permissible operations on it.
23.	Demonstrate the B Tree and perform all the permissible operations on it.
24.	Demonstrate the B* Tree and perform all the permissible operations on it.
25.	Create Adjacency Matrix and Adjacency List for a defined Graph
26.	Perform Depth First Search & Breadth First Search on a Graph
27.	Generate all pair Shortest Paths in a Graph
28.	Compute Shortest Path between any two vertices in a Graph
29.	<p>Perform the following:</p> <ul style="list-style-type: none"> i. Dijkstra Algorithm ii. Bellman-Ford Algorithm

MSc. IT Syllabus – Department of Computer Science, IUST

	iii. Floyd Warshall Algorithm
30.	Perform the Following in a Spanning Tree: i. Prim's Algorithm ii. Kruskal's Algorithm



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Introduction to Python

Semester: 2nd

Credits: 05

Pre Requisite: MIT501C, MIT502C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

Course Code: MIT552C

Paper Type: Core

Max Marks:: 125

Co-Requisite: MIT551C

COURSE OBJECTIVES:

- *To learn the essential concepts of Python programming*
- *The course aims at equipping participants to be able to use python programming for solving data science problems.*
- *To perform data visualization and presentation of preprocessed data.*
- *To understand the advantage of using Python libraries for data analysis and implementing Machine Learning models*

COURSE CONTENT:

UNIT I

Introduction to Python and its history, setting up Python development environment (IDEs, tools), Python syntax, indentation, and coding conventions.

Variables, identifiers, and reserved keywords; Numeric data types (int, float) and arithmetic operations; Conditional statements (if, elif, else), Looping constructs (for, while), Using break and continue in loops.

Python Data structures - lists, tuples, dictionaries, sets; List manipulation – indexing and slicing. Dictionary manipulation; String, list and dictionary in-built functions.

UNIT II

Introduction to function - Defining and calling functions, Function parameters, return values and scope, Lambda functions.

File Handling - Reading and writing files in Python, Working with CSV and JSON files.

Object Oriented Programming – Introduction to OOPs concepts like classes, objects; inheritance and polymorphism.

UNIT III

Numpy: NdArray and its operation, Indexing, slicing, broadcasting. Boolean arrays, shape manipulations. Pandas - pandas dataframes, reading data from files, Data preparation and preprocessing. Data visualization on dataset using matplotlib / seaborn library: Different types of plots - Scatter plot, Line plot, Bar plot, Histogram, etc.

UNIT IV

Building RESTful APIs with Python, Overview of web development frameworks (Flask/Django); Working with databases (SQL, SQLite); Web scraping with Python.

Capstone Projects : Students will work on a real-world Python project of their choice.

Project topics can include web applications, data analysis, machine learning, automation, etc.

COURSE OUTCOMES:

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

- *Read, write, and execute simple Python programs.*
- *Write simple Python programs for solving problems.*
- *Decompose a Python program into functions, lists etc.*
- *Read and write data from/to files in Python Programs*
- *Perform Exploratory Data Analysis*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. Mastering Python for data science, Samir Madhavan
2. Kenneth A. Lambert, The Fundamentals of Python: First Programs, Cengage Learning, ISBN: 978-1111822705.
3. Introduction to Machine Learning with Python, Andreas C. Mueller

References:

1. Machine Learning using Python, U Dinesh Kumar Manaranjan Pradhan
2. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.
3. Jake VanderPlas "Python Data Science Handbook" O'Reilly Publications.
4. Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc."
5. Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

List of Practicals:

LAB JOURNAL_MIT552C	
S. No.	TITLE
1.	Write a program to demonstrate different number of data types in python.
2.	Write a program to perform different arithmetic operations on numbers in python.
3.	Write a program to create, concatenate and print a string and accessing substring from a given string.
4.	Write a python script to print the current date in following format "Sun May 29 02:26:23 IST 2017"
5.	Write a python program to create, append and remove lists in python.
6.	Write a program to demonstrate working with tuples in python
7.	Write a program to demonstrate working with dictionaries in python
8.	Write a python program to find largest of three numbers
9.	Write a python program to convert temperature to and from Celsius to Fahrenheit.
10.	Write a python program to construct the following pattern using nested for loop:
11.	Write a python program to print prime numbers less than 20:
12.	Write a python program to find factorial of a number using recursion:
13.	Write a python program to that accepts length of three sides of a triangle as inputs. The program should indicate whether or not the triangle is a right-angled triangle (use Pythagorean theorem):
14.	Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15.	Write a python program to define a module and import a specific function in that module to another program.
16.	Write a program to demonstrate the use of various operators in python.
17.	WAP to demonstrate the implementation of various control flow structures in python.
18.	Write a Program to demonstrate the use of various looping structures in python.
19.	Write a Program to demonstrate the implementation of functions, modules and packages in python
20.	Write a Program to demonstrate various operations associated with files(Creating, opening, reading, writing, delete, update)
21.	Write a Program to demonstrate the implementation of Lists and its associated operations.
22.	Write a Program to demonstrate the implementation of Tuples and its associated operations.
23.	Write a Program to demonstrate the implementation of Dictionary and its associated operations.

MSc. IT Syllabus – Department of Computer Science, IUST

24.	Write a Program to demonstrate the implementation of Strings and its associated operations.
25.	Write a Program to demonstrate the implementation of Sets and its associated operations.
26.	Write a Program to demonstrate the use of Numpy to perform various operations (Creating an Array (1-D,2-D,3-D), Adding, subtracting, Multiplying, Indexing, Slicing etc).
27.	Using a numpy module create an array and check the following: i. Type of array ii. Axes of array iii. Shape of array iv. Type of elements in array
28.	Using a Numpy module create array and check the following: i. List with type float ii. 3*4 array with all zeros
29.	Write a program to demonstrate the implementation of various Linear Algebra and related operations using python.
30.	Write a program to demonstrate the use of Pandas Library in Python
31.	Write a Pandas program to create and display a one-dimensional array-like object containing an array of data using Pandas module
32.	Write a pandas program to convert a pandas module series to python list and its type
33.	Write a pandas program to add, subtract, multiple and divide two pandas series
34.	Write a pandas program to convert a Numpy array to a pandas series.
35.	Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.
36.	Write a Pandas program to change the name 'James' to 'Suresh' in name column of the DataFrame.
37.	Write a Pandas program to insert a new column in existing DataFrame
38.	Write a Pandas program to get list from DataFrame column headers.
39.	Write a Pandas program to join the two given dataframes along rows and assign all data.
40.	Write a Pandas program to read excel and CSV file to perform operations: drop null values, locate null values, fill null values, display file contents, etc.
41.	Write a python program to demonstrate Data visualization on dataset using matplotlib and seaborn libraries: Scatter plot, Line plot, Bar plot, Histogram, Box plot, Pair plot.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Data Communication & Computer Networks	Course Code:	MIT553C
Semester:	2 nd	Paper Type:	Core
Credits:	04	Max Marks:	100
Pre Requisite:	MIT526E, MIT541A, MIT527E	Co-Requisite:	...
Marks Distribution:	(Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)		

COURSE OBJECTIVES:

- *To understand networking concepts and basic communication model.*
- *To understand network architectures and components required for data communication.*
- *To analyze the function and design strategy of physical, data link, network layer and transport layer*
- *To acquire basic knowledge of various application protocol for internet security issues and services.*

COURSE CONTENT

UNIT I

Principles of Data Communication, Data Communication Model & Tasks, Concept of Bandwidth & Channel Capacity, Nyquist's Law & Shannon's Law for Quantification of Capacity, Data Rate Versus Baud Rate, Sampling & its types, Nyquist Criterion for Sampling, Data Transmission Concepts, Characteristics of Signals: Digital Signals, Analog Signals (Amplitude, Frequency, Period, Wavelength, Signal to Noise Ratio).

UNIT II

Data Encoding: NRZ-L, NRZ-I Encoding, Multilevel Binary & Biphase Encoding Techniques & their implementations. ASK, FSK, PSK & QPSK. PCM: Sampling & Quantization & Modulation, Delta Modulation, Amplitude Modulation. Reliable Transmission of Data: Asynchronous & Synchronous Transmission, Multiplexing: Concepts & Types (FDM, Synchronous & Statistical TDM). Error Detection: Parity Based, CRC-Based, Checksum Based. Error Control & Recovery Techniques, Concept of ARQ Standard & its Versions.

UNIT III

Goals & Applications of Computer Network. OSI model. TCP/IP protocol suite. LAN: Ethernet & Token Ring. WAN: Concept, Characteristics, & Architecture: WAN subnet. Internetworking Concept & Architectural Model, Connection Oriented & Connectionless Approaches, Packet switching & Circuit switching. Concept of Autonomous Systems. Internet Layer Protocols: IP (Addressing: Classful & classless IP Addressing, IP Multicasting, Routing, Fragmentation & Reassembly).

UNIT IV

ICMP, ARP, RARP. Routing Protocols: Interior (OSPF), Exterior (BGP). Transport Layer Protocols: TCP & UDP. Socket API for Network Programming: Concept of Port & Sockets. Basic Server Architectures. TCP Server & Client. UDP Server & Client. Network Security: Principles of Security, Firewalls & their Components, Encryption Techniques & examples of Encryption Standard (DES, AES & RSA).

COURSE OUTCOMES:

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

- *Describe how to connect machines in a network.*
- *Describe data communication between machines at various locations.*
- *Familiarize with LAN connecting devices and Network Layer.*
- *Learn different Network Layer Protocols.*
- *Understand the concept of Domain Name System (DNS).*

Text Books:

1. Computer Networks ,A.S. Tanenbaum, Pearson Education
2. Data Communications & Networking ,Forouzan, Tata McGraw Hill Company

School of Engineering & Technology (SoE&T),
Islamic University of Science & Technology (IUST),

References:

1. Computer Network, S.S.Shinde, New Age International Publisher.
2. Data & Computer Communication , Shashi Banzal ,Firewall media
3. Data & Computer communication, William Stallings, Pearson



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Operating System	Course Code:	MIT554C
Semester:	2 nd	Paper Type:	Core
Credits:	04	Max Marks:	100
Pre Requisite:	MIT501C, MIT504C, MIT502C	Co-Requisite:	MIT551C
Marks Distribution:	(Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)		

COURSE OBJECTIVES:

- *To learn the fundamentals of Operating Systems.*
- *To learn the mechanisms of OS to handle processes and threads and their communication*
- *To learn the mechanisms involved in memory management I/O management and file management in OS*
- *To gain knowledge on Mutual exclusion algorithms, deadlock detection algorithms*
- *To know the components and management aspects of concurrency management*
- *To implement the concepts learnt through case studies of different operating systems.*

COURSE CONTENT:

UNIT I

Overview of an Operating system, Functions of Operating System. Process and thread Management Concepts: Process Scheduling, Scheduling Criteria, Scheduling algorithms, Thread Scheduling. Process Synchronization, Semaphores, Critical Section and Monitors, Inter-process Communication, shared memory model and message passing. Deadlocks: Concept of Deadlock, Deadlock prevention, avoidance, detection and Deadlock recovery.

UNIT II

Memory Management: Linking, Loading, Memory Allocation, Design Issues & Problems, Fragmentation, Compaction, Memory Management Unit, Paging, Segmentation, Virtual Memory, Demand Paging. Page Replacement Algorithms. Allocation Algorithms, Thrashing.

UNIT III

File Management: File Structure, File Protection, File System Implementation, Directory Structure, Free Space Management, Allocation Methods, Efficiency & Protection. Disk Management: Disk Structure, Disk Scheduling Algorithm, Disk Management, Swap Space concept & Management, RAID Structure, Disk Performance issues.

UNIT IV

Multiprocessor Systems: Types of Multiprocessor Operating Systems, Functions & Requirements, Design & Implementation Issues. Distributed Operating System: Difference between Distributed & Centralized Operating System, Advantages & Disadvantages of Distributed Operating System,

Hardware & Software Concepts, Loosely Coupled Systems, Types of Distributed Operating System.

COURSE OUTCOMES:

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

- *Master understanding of design issues associated with operating systems.*
- *Master various process management concepts including scheduling, synchronization, and deadlocks.*
- *Be familiar with various types of operating systems including UNIX.*

Text Books:

1. Tanenbaum, A.S., "Modern Operating System", PHI
2. Peterson, J.L. Abraham, Silberschatz, "Operating System Concepts", Addison Wesley

MSc. IT Syllabus – Department of Computer Science, IUST

References:

1. Dietel H.M. "An Introduction To Operating System", Addison Wesley
2. Karnetkar, "UNIX Shell Programming", BPB
3. W.Stallings, "Operating systems"
4. Dhamdhere, "An Operating System –Design & principles"
5. Madnick E, Donovan J, "Operating Systems", TMH
6. Marko Gergent, "Learning android", O’rielly



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Web Mining	Course Code:	MIT561E
Semester:	2 nd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT502C, MIT526E, MIT512E, MIT528E	Co-Requisite:	MIT552C, MIT553C
Marks Distribution:	(Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)		

COURSE OBJECTIVES:

- *To acquaint the students about the types of data which exists on web sites/portals.*
- *To provide students will be able to know various techniques which are using in mining of data from the web portals.*

COURSE CONTENT:

UNIT I

Introduction to Web Data Mining: Data Mining Foundations, Introduction – World Wide Web (WWW), A Brief History of the Web and the Internet, Web Data Mining-Data Mining, Web Mining. Data Mining Foundations – Association Rules and Sequential Patterns – Basic Concepts of Association Rules, Apriori Algorithm- Frequent Item set Generation, Association Rule Generation.

UNIT II

Supervised Learning: Supervised Learning – Basic Concepts, Decision Tree Induction – Learning Algorithm, Impurity Function, Handling of Continuous Attributes, Classifier Evaluation, Rule Induction – Sequential Covering, Rule Learning, Classification Based on Associations, Naïve Bayesian Classification, Naïve Bayesian Text Classification – Probabilistic Framework, Naïve Bayesian Model.

Unsupervised Learning – Basic Concepts, K-means Clustering – K-means Algorithm, Representation of Clusters, Hierarchical Clustering – Single link method, Complete link Method, Average link method, Strength and Weakness.

UNIT III

Information Retrieval and Web Search: Basic Concepts of Information Retrieval, Information Retrieval Methods – Boolean Model, Vector Space Model and Statistical Language Model, Relevance Feedback, Evaluation Measures, Text and Web Page Preprocessing. Link Analysis and Web Crawling: Link Analysis – Social Network Analysis, Co-Citation and Bibliographic Coupling, Page Rank Algorithm, HITS Algorithm

COURSE OUTCOMES:

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

- *Understand different kinds of web mining techniques.*
- *Understand the various techniques for web content mining.*
- *Understand various concepts related to NLP.*
- *Understand extraction of data from social networks.*
- *Explain semantic web and web spam analysis.*

Text Books:

1. George C, Marcus J. H, James A. M. McHugh, Jason T.L. Wang, “ Mining the World Wide Web: An Information Search Approach”, Kulwer Academic Publishers.

References:

1. Soumen Chakrabarti, “ Mining the Web: Discovering Knowledge from Hypertext Data”, Morgan Kufman Publishers.

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Quantitative Techniques

Semester: 2nd

Credits: 03

Pre Requisite: MIT503C

Course Code: MIT562E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT551C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To identify and develop operational research models from the verbal description of the real system.
- To understand the mathematical tools that are needed to solve optimization problems.
- To define and formulate linear programming problems and appreciate their limitations.
- To understand how to solve linear programming problems using appropriate techniques and optimization solvers, interpret the results obtained and translate solutions into directives for action.
- To conduct and interpret post-optimal and sensitivity analysis and explain the primal-dual relationship.
- To develop mathematical skills to analyze and solve integer programming and network models arising from a wide range of applications.

COURSE CONTENT:

UNIT I

Linear Programming: L P formulation, Graphical methods for LPP with 2 variables, Simplex Algorithm for LPP, Duality theorem in linear programming & applications. Transportation problem: Formulation, methods of selecting initial feasible solutions, Degeneracy & resolution. Assignment problem: Hungarian Method for solving Assignment Problems, Balanced & Unbalanced problems & resolution.

UNIT II

Network Analysis: Shortest routes, Enumeration & applications. Critical Path Method (CPM) & Programme Evaluation and Review Technique (PERT): Use & design of CPM & PERT. Critical Path calculation, Float & its Types, Crashing in Project Management.

Sequencing problems, Johnsons algorithm for processing m jobs through 2, 3 & ' n ' machines.

UNIT III

Replacement & Sequencing models: Replacement of items that fail & deteriorate. Group & individual replacement.

Game theory: definition & explanation, saddle points, Dominance mixed strategies, games without saddle points, $2 \times N$ games.

Dynamic Programming: Characteristics of dynamic programming problem, Bellman's optimality principles, dynamic programming under certainty, shortest route problem,

COURSE OUTCOMES:

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

- Formulate and solve linear Programming Problems
- Determine the optimum solution to constrained and unconstrained
- Apply dynamic programming principle to Linear programming problems.
- Determine the integer solutions to Linear Programming Problems

Text Books:

1. N.D.Vohra, "Quantitative Techniques in management", TMH
2. Hamdy A. Taha, "Operations Research: An Introduction", Pearson

MSc. IT Syllabus – Department of Computer Science, IUST

References:

1. Sharma J. K., "Operations Research: Theory & Applications", Macmillan India
2. Gross Donald, "Fundamentals of Queuing Theory", 3rd Ed., John Wiley
3. Mokhtar S. Bazaraa, "Linear Programming & Network Flows", John Wiley
4. Hiller Lieberman, "Introduction to Operations Research", TMH
5. Laudon, "Decision Support Systems", PHI



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Microprocessor & Assembly Language Programming	Course Code:	MIT563E
Semester:	2 nd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT501C, MIT504C, MIT511E	Co-Requisite:	MIT561E
Marks Distribution:	(Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)		

COURSE OBJECTIVES:

- *To understand the principles of instruction set architecture and assembly language programming.*
- *To understand basic procedures of how a compiler translates C/C++ code to assembly language and perform simple optimizations.*
- *To understand basic principles of interrupt/exception handling.*
- *To explore in detail a simple hardware CPU implementation that supports a small instruction subset.*

COURSE CONTENT:

UNIT I

Software Model of 8088 / 8086 Microprocessor, Memory Add. Space & Data Organization, Data Types, Segment Registers, Memory Segmentation Dedicated, reserved & general use of Memory, generating an Memory Address, Pin-out diagram of 8086 Microprocessor.

UNIT II

The Microcomputer Organization, Assembly Language Programming Development on PC, Instruction Set, Addressing Modes, 8086 Instruction set, Integer Instructions & Computations, Data Transfer, Arithmetic, Logic Shift, Rotate Instruction, Flag Control, Compare, Control Flow & Jump, Subroutine & Subroutine Handling Instructions, Loop & Loop Handling, String & String Handling Instructions. Statement Syntax for a source Program, Assembler Directives, Assembling, Linking, Loading & executing a run Module.

UNIT III

Isolated I / O, Memory Mapped I/O, DMA Controller, Programming Communication Interfaces Controller. Interfacing I/O devices to microprocessor, programmable peripheral interface. Interrupt, Mechanism, Types & Priority, Interrupt Vector table, Real Mode. 8086 / 8088 Microprocessors, 8086 / 8088 Microprocessor's Minimum Mode, Maximum Mode Systems, Bus Cycle & Unit States, Memory Control Signals, Read & Write Bus Cycles, Memory Interface Circuits.

COURSE OUTCOMES:

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

- *Describe the architecture of 8086*
- *Develop simple program using 8086*
- *Describe the basic peripheral devices and its applications*
- *Differentiate various microprocessor architectures*

Text Books:

1. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications (Prentice Hall international editions) Walter A. Triebel; Avtar Singh

References:

1. DOUGLAS HALL "Microprocessors & Interfacing" Tata McGrawHill.
2. LIU, GIBSON et al "Microcomputer system The 8086/8088 Family" PHI.

MSc. IT Syllabus – Department of Computer Science, IUST

3. PAL CHAUDHURI “Computer Organization & Design” PHI.
4. MORRIS MANO “Computer System Architecture” Pearson Education.
5. GILMORE “Microprocessors” Wiley/ Tata McGraw Hill.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Internet of Things	Course Code:	MIT564E
Semester:	2 nd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT501C, MIT526E	Co-Requisite:	MIT553C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *The main objective of this course is to make student familiar with the concepts of Internet of Things and existing technologies based on it.*
- *It will also provide the way to identify the problems which might be solved by IoT with feasible solution.*

COURSE CONTENT:

UNIT I

Introduction and Concepts: Components of Internet of Things (IoT), Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels and Deployment Templates. IoT Architecture: Reference Model, SOA-Based, API-Based

UNIT II

IoT Devices & Objects: Transducers, Actuators, Sensors, Smart Objects, Embedded Devices, Arduino, Raspberry Pi, RFID.

Introduction Communication Technologies: M2M, D2D, D2Gateway, Difference between IoT & M2M.

Introduction to Computing Technologies: Edge Computing, Fog Computing, Cloud Computing with respect to IoT

UNIT III

Introduction to Protocols: IoT Protocol Stack, IPv6. Application Layer Protocols: CoAP, MQTT, XMPP, AMQP. Data Link Protocols: IEEE 802.15.4e, IEEE 802.11 ah, Bluetooth Low Energy, Zigbee, LoRaWAN. Network Layer Routing/Encapsulation Protocols: RPL, CORPL, CARP, 6LoWPAN. Session Layer Protocols: MQTT, SMQTT, AMQP, CoAP, XMPP

COURSE OUTCOMES:

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

- *Get the concept of basic concepts of Internet of Things.*
- *Understand machine to machine communication systems.*
- *Analyze the real life problems for providing technology based solution.*
- *Identify the problems for providing technology based solutions and have skills on the Design of IoT based Applications.*

Text Books:

1. Bahga, A, Madiseti, V (2015), "Internet of Things-A Hands-on Approach", Universities Press (India).
2. Buyyaand, R, Dastjerdi,A.V(2016) , "Internet of Things: Principles and Pradigms", Cloud Computing and Distributed Systems (Clouds) Laboratory, ManjaSoft Pvt Ltd., Australia.
3. FeiHu, "Security and Privacy in Internet of Things (IoTs) Models, Algorithms and Implementations", CRC Press.

MSc. IT Syllabus – Department of Computer Science, IUST

References:

1. Kellmerit. D, Obodovski. D (2013) ,“ The Silent Intelligent-The Internet of Things”, Published by DND Ventures LLC, 1st Edition.
2. Hersent. O, Boswarthick . D, Elloumi . O (2017),“The Internet of Things Key Applications and Protocols”, Published by Wiley, Reprint.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Open Source Technologies

Semester: 2nd

Credits: 03

Pre Requisite: MIT512E, MIT526E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT576E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT552C

COURSE OBJECTIVES:

- To learn about Linux Administration.
- To learn PHP and its database connectivity.
- To learn how to create Websites with PHP MY-SQL.
- To learn Content Management System. And Introduction to No-Sql.

COURSE CONTENT:

UNIT I

Introduction to Linux – Architecture, Installation, Configuration, Package Management, Basic Linux commands - file system commands, vim. Apache Http Server – installation, basic configuration, setting up virtual hosts, installation of modules, etc

UNIT II

Introduction to PHP - variables, data types, basic constructs, loops, functions, classes. Installation of third party php modules. Introduction to MySQL database. PHP-MySQL connectivity. Introduction to HTTP – HTTP methods, headers, cookies, etc. Creating a basic website with PHP & MySQL.

UNIT III

Content management system based on php & mysql e.g. Wordpress & Drupal – installation, configuration. Introduction to NoSQL – installation & configuration of Mongoddb. Basic select, insert, update & delete operations, aggregate operations. Backup & restoration.

COURSE OUTCOMES:

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

- Explore different open source technology like Linux, PHP & MySQL with different packages.
- Execute Linux commands for programming.
- Execute programs of PHP with MySQL connection

Text Books:

1. Core Java Volume 1-Fundamentals by Cay S.horstman 10th edition Publisher: Prentice Hall

References:

1. Java the complete reference by Herbert Schield 10th edition Publisher:Tata Mc Graw Hill
2. Java How to program Dietel and Dietel

List of Practicals:

LAB JOURNAL_ MIT576E	
S. No.	TITLE
1.	Linux Architecture & Installation Install a Linux distribution (e.g., Ubuntu, CentOS). Explore the Linux file system.
2.	Linux Package Management Use package managers like apt or yum to install and update software packages.
3.	Basic Linux Commands & Vim Practice basic file system commands like ls, cd, mkdir, etc. Learn basic Vim commands for text editing.

MSc. IT Syllabus – Department of Computer Science, IUST

4.	Apache Http Server Install Apache HTTP Server. Configure basic settings like port number and document root.
5.	Apache Virtual Hosts Set up virtual hosts for multiple websites on a single server.
6.	Introduction to PHP Learn about PHP variables, data types, and basic constructs. Practice writing loops, functions, and classes in PHP.
7.	Installation of PHP Modules Install third-party PHP modules (e.g., PHP-GD for image processing).
8.	Introduction to MySQL Database Install MySQL Database Server. Perform basic operations like creating databases and tables.
9.	PHP-MySQL Connectivity Write PHP scripts to connect to and interact with a MySQL database.
10.	Introduction to HTTP Learn about HTTP methods, headers, and cookies. Practice sending HTTP requests and handling responses.
11.	Creating a Basic Website with PHP & MySQL Develop a simple website using PHP for server-side scripting and MySQL for database storage.
12.	Content Management Systems (CMS) Install and configure a CMS like WordPress or Drupal. Explore the admin interface and create basic content.
13.	Introduction to NoSQL Install and configure MongoDB. Practice basic operations like inserting, updating, and deleting documents.
14.	NoSQL Aggregation Operations Learn about MongoDB's aggregation framework. Practice using aggregation operations like \$match, \$group, and \$sort.
15.	Backup & Restoration Perform backups of your MySQL and MongoDB databases. Practice restoring databases from backups.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Computer Graphics
Semester: 2nd
Credits: 03
Pre Requisite: MIT501C, MIT503C

Course Code: MIT577E
Paper Type: DCE
Max Marks: 75
Co-Requisite: MIT551C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- To explain basic primitives like Point, Line, Circle, Ellipse etc. with their mathematical equations coupled with the geometrical interpretations.
- To explain process of displaying continuous primitives on Discrete Display Devices.
- To explain few transformations like Translation, Rotation, Scaling and composition of these transformations.
- To explain Projections from higher dimensional setup to lower dimensions.
- To explain drawing of complex scenes using Splines coupled with understanding of spline specifications and few variants of splines besides familiarizing about Beziers.
- To acquaint students about Concepts of displaying structures with no inherent regular geometry using Fractals.

COURSE CONTENT:

UNIT I

An Introduction Graphics System: Computer Graphics & Its Types, Application of computer graphics, Graphics Systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Display Buffer, Concept of Double Buffering & Segmentation of Display Buffer. Use of Lookup tables. "Introduction to Color models (RGB, CMY, and HSV)."

UNIT II

Output Primitives & Attributes of Output Primitives: Output Primitives Points & Lines, Line Drawing Algorithms, Circle Generating Algorithms, Scan-Line Polygon Fill Algorithm, Inside-Outside tests, Boundary-Fill Algorithm, Flood Fill Algorithm, Cell Array, Character Generation, Attributes of Output Primitives: Line attributes, Color & Grayscale Levels, Area fill Attributes, Character Attributes, Bundled Attributes. Anti-aliasing. Two-dimensional Geometric Transformations: Basic Transformations, Matrix Representations & Homogeneous Coordinates, Composite Transformations, Reflection & Shearing.

UNIT III

Two-Dimension Viewing: The viewing Pipeline, Window to view port coordinate transformation, Clipping Operations, Point Clipping, Line Clipping, Polygon Clipping, Text Clipping, Exterior Clipping.

Three-Dimensional Concepts: Three Dimensional Display Methods, 3D Transformations, Parallel Projection & Perspective Projection

Curves & Surfaces, Splines, Spline specification, Interpolated & Approximated Splines. Bezier Splines, Bezier Curves, Cubic Bezier Curves, Bezier Surfaces. B-Splines curves & surfaces. Fractals - Fractal Generation Procedure.

COURSE OUTCOMES:

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

- Understand concepts of computer graphics.
- Understand 2D and 3D transformation, clipping, splines, objects modeling, colour modeling, lighting, textures, visible surface detection.
- Understand algorithms to design and create computer graphics scenes.

Text Books:

1. Hughes, John F., et al. Computer graphics: principles and practice. Pearson Education, 2014.

MSc. IT Syllabus – Department of Computer Science, IUST

References:

1. W.M.Newman & Sproull. "Principles of interactive Computer Graphics" ,TMH
2. Steven Harrington." Computer Graphics a Programming Approach" McGraw Hill.
3. Plastock & Kelley. "Schaums outline of theory & problems of computer Graphics"
4. David F Rogers & J Alan Adams. "Procedural Elements of Computer Graphics" McGraw Hill
5. David F Rogers & J Alan Adams. "Mathematical Elements of Computer Graphics" McGraw Hill

List of Practicals:

LAB JOURNAL_ MIT577E	
S. No.	TITLE
1.	Write a program in C/ C++ for displaying a line segment using the DDA algorithm.
2.	Write a program in C/ C++ for computing distance between two given points.
3.	Write a program in C/ C++ for finding midpoint between two given points in 2 dimensions
4.	Write a program in C/ C++ for displaying a line segment using Bresenham's Line drawing algorithms.
5.	Write a C++ to check whether a given line segment passes through a given set of points.
6.	Write a C++ program to find neighbors of a given point on circle with radius r and center (0, 0)
7.	Write a C++ program to find neighbors of a given point on circle with radius r and center (xc, yc)
8.	Write a program in C/ C++ to find whether a given point lies Inside, Outside or on the boundary of a given circle with center (xc, yc) and radius r.
9.	Write a program in C/ C++ to implement Mid-Point Circle Drawing Algorithm.
10.	Use polar coordinate system to display a circle with radius r and centre (xc, yc)
11.	Write a program in C/ C++ to implement a midpoint Ellipse drawing algorithm.
12.	Write a program in C/ C++ to do following basic 2-dimensional transformations <ol style="list-style-type: none">i. Translationii. Rotationiii. Scaling
13.	Write a program in C/ C++ to implement Basic/ Composite 2-dimensional transformations using matrix multiplication.
14.	Write a program in C/ C++ to draw a rectangular window of any arbitrary size with dimensions (Xwmin, Ywmin, Xwmax, Ywmax)
15.	Write a program in C/ C++ to draw a rectangular viewport window of any arbitrary size with dimensions (Xvmin, Yvmin, Xvmax, Yvmax)
16.	Write a program in C/ C++ to implement Window to Viewport transformation.
17.	Write a program in C/ C++ to implement flood fill algorithm
18.	Write a program in C/ C++ to fill polygon using flood-fill algorithm
19.	Write a program in C/ C++ to draw a rectangle
20.	Write a program in C/ C++ to drawing circle
21.	Write a program in C/ C++ to draw a circle without floating point arithmetic
22.	Write a program in C/ C++ to rotate an object by a given angle about a given point
23.	Write a program in C/ C++ for point clipping Algorithm
24.	Write a program in C/ C++ to implement Cohen Sutherland algorithm for line clipping.
25.	Write a program in C/ C++ for implementing Sutherland–Hodgman algorithm for polygon clipping

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Numerical & Statistical Computing	Course Code:	MIT578E
Semester:	2 nd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT501C, MIT503C	Co-Requisite:	MIT552C, MIT562E
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- To explain concept of Polynomials, Equations, Zeros and Roots.
- To explain few iterative techniques for computation of roots together with their geometrical interpretation and implementation in C.
- To explain solving system of equation using known methods along with their geometrical interpretation and implementation in c.
- To explain concepts of Curve Fitting with specific focus on Lagrange's interpolating Polynomial with a prior explanation of using linear and nonlinear curves for curve fitting.
- To explain few statistical concepts measures of central tendency, Dispersion etc.
- To explain concept of Hypothesis, Null hypothesis besides giving students knowledge about few tests which include T test, F test etc.
- To make students ready for courses like Artificial Intelligence and Machine Learning by giving them prior knowledge about Linear Algebra, Probability and statistics.

COURSE CONTENT:

UNIT I

Approximations & Errors – Types of Programming Errors, Data Errors, Computer & Arithmetic Errors, Round Off and Truncation Errors. Accuracy and Precision, Measures of Accuracy, Error Propagation. Non-Linear Equations, Types of Methods to find solutions to nonlinear equations, Algorithms to Compute Roots of Equation – Methods of Tabulation or Brute Force Method, Method of Bisection, Secant Method, Newton-Raphson Method, Method for False Position. Derivation of mathematical formulas, geometric interpretation and implementation of these methods.

UNIT II

Linear Equations, Types of Methods to find solutions to linear equations. Algorithms to Solve Linear Algebraic Equations: Gauss Elimination, Gauss Jordan, Gauss Seidel, L.U. Decomposition, Lagrange Interpolated Polynomial, Newton's Methods of INTERPOLATION – Forward difference, Backward difference. Derivation of mathematical formulas and implementation of these methods.

UNIT III

Differential Equations – Concepts and Terminology, Algorithms to solve Ordinary Differential Equations – Euler Method and Modification. The trapezoidal Rule, Simpson's Rule. 4th order R-K Method. Derivation of mathematical formulas and implementation of these methods.

COURSE OUTCOMES:

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

- Understand the iterative methods to find solution of polynomial and transcendental equations.
- Discuss methods of interpolation and curve fitting.
- Find the solution of linear equations using matrices.

Text Books:

1. Rajaraman, Vaidyeswaran. Computer oriented numerical methods. PHI Learning Pvt. Ltd., 1993.

References:

1. S.C.Chapra and R.P.Canale: "Numerical methods for Engineering". Tata McGraw Hill.

MSc. IT Syllabus – Department of Computer Science, IUST

2. Krishenmurty & Sen : “Numerical Algorithms”
3. V. Rajaraman “Computer oriented numerical methods.” Prentice Hall of India.
4. McCalla, Thomas Richard: “Introduction to Numerical Methods & FORTRAN Programming”, John Wiley & Sons, Inc.
5. Grewal, B. S.: “Higher Engineering Mathematics”, Hindustan Offset Problems Series.
6. “SCHAUM’S Solved Problems Series”.
7. Sharma, K. D.: “Programming in Fortran”.
8. Jain, M. K., Iyengav, S. R. K., Jain, R. K.: “Numerical Methods for Scientific & Engineering Computation”+, Wiley Eastern Ltd, New Delhi.

List of Practicals:

LAB JOURNAL_ MIT578E	
S. No.	TITLE
1.	Write a program in C/ C++ to find real roots of nonlinear equation using Bisection Method
2.	Write a program in C/ C++ to find real roots of nonlinear equation using Regula Falsi or False Position Method
3.	Write a program in C/ C++ to find real roots of nonlinear equation using Newton Raphson Method
4.	Write a program in C/ C++ to find real roots of nonlinear equation using Secant Method
5.	Write a program in C/ C++ which solve systems of linear equation using Gauss Elimination Method
6.	Write a program in C/ C++ which solve systems of linear equation using Gauss Jordan Method
7.	Write a program in C/ C++ to find inverse of matrix using Gauss Jordan Method
8.	Write a program in C/ C++ to find largest or dominant Eigenvalue and corresponding Eigenvector using Power Method
9.	Write a program in C/ C++ to solve a system of linear equations using the Jacobi Iteration Method.
10.	Write a program in C/ C++ for solving systems of linear equations using Gauss Seidel iterative method
11.	Write a program in C/ C++ to generate forward difference table using discreetly defined Function
12.	Write a program in C/ C++ to generate backward difference table using discreetly defined function
13.	Write a program in C/ C++ which implements Lagrange Interpolation
14.	Write a program in C/ C++ which implements Linear Interpolation Method
15.	Write a program in C/ C++ which reads n data points from user and then implement linear regression on the given data.
16.	Write a program in C/ C++ which implements pseudocode for Curve Fitting of Type $y=aX^b$
17.	Write a program in C/ C++ which implements pseudocode for Curve Fitting of Type $y=ab^x$
18.	Write a program in C/ C++ to find derivatives using Newton's forward difference formula.
19.	Write a program in C/ C++ to find derivatives using Newton's backward difference formula.
20.	Write a program in C/ C++ to find numerical integration using Trapezoidal Rule or Method.
21.	Write a C program to implement R. K. 2nd order method for computing integral of a given function.
22.	Write a C program to implement R. K. 4th order method for computing integral of a given function.

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Pervasive Computing	Course Code:	MIT579E
Semester:	2 nd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT501C, MIT503C, MIT512E	Co-Requisite:	MIT551C, MIT552C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- To introduce students to the fundamental concepts and principles of pervasive computing, including ubiquitous computing, ambient intelligence, Internet of Things (IoT), and wearable computing.
- To familiarize students with the diverse range of technologies and devices used in pervasive computing systems, such as sensors, actuators, RFID tags, smart appliances, and mobile devices.
- To provide an overview of pervasive computing architectures, communication protocols, and middleware used to enable seamless connectivity and interaction between heterogeneous devices and systems.
- To explore techniques for sensing and context awareness, including sensor fusion, context modeling, and context reasoning, to enable systems to adapt to users' changing contexts and preferences.
- To emphasize the importance of user-centered design principles in pervasive computing applications, considering factors such as usability, accessibility, privacy, and security.
- To provide hands-on experience in developing pervasive computing applications using appropriate programming languages, frameworks, and development tools. This may involve mobile app development, web-based applications, or embedded systems programming.

COURSE CONTENT:

UNIT I

Technologies : Past, Present, Future , Pervasive Computing , The pervasive computing market, Device Technology : Hardware , Human-machine interfaces , Biometrics , Operating Systems , Java for Pervasive devices , Device Connectivity : Protocols , Security , Device Management , Web Application Concepts : History of World wide Web . World Wide Web Architecture, Protocols, Transcoding, Client Authentication via the Internet.

WAP: Introduction , Components of the WAP architecture , WAP infrastructure , WAP Security Issues , Wireless Markup Language , WAP push , Products , i-mode , Voice Technology.

UNIT II

Wireless Markup Language , WAP push , Products , i-mode , Voice Technology : Basics of Speech Recognition , voice standards , speech applications , speech and pervasive computing, security, Personal Digital assistants : History , Device Categories , PDA Operating Systems , Device Characteristics , Software Components , Standards , Mobile Applications , PDA browsers.

UNIT III

Architecture : Server Side Programming in Java : J2EE and overview , Servlets, Enterprise Java Beans , Java Server Pages , Extensible Markup Language , Web services , Model-View-Controller Pattern, Pervasive web application architecture : Background , scalability and availability ,

Smart Card-based authentication via the Internet, Ordering goods, Access via WAP : WAP functionality , Implementation , Access from Personal Digital Assistants: Extending the example application to personal digital assistants. Implementation for synchronized devices, for intermittently connected devices, for connected devices. Access via voice.

COURSE OUTCOMES:

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

- Design intelligent machines/systems which act rationally and take the right decision at the right time.
- Understand pervasive computing principles, including ubiquitous computing, ambient intelligence, and the Internet of Things (IoT).

MSc. IT Syllabus – Department of Computer Science, IUST

- *Familiar themselves with the diverse range of technologies and devices used in pervasive computing systems, including sensors, actuators, RFID tags, smart appliances, and mobile devices.*
- *To design architectures and protocols for pervasive computing systems, considering factors such as scalability, interoperability, and resource constraints.*
- *To demonstrate proficiency in sensing and context awareness techniques*

Text Books:

1. Pervasive Computing: Technology And Architecture Of Mobile Internet Applications. First Edition, BURKHARDT, Pearson India

References:

1. JochenBurkhardt, Dr. Horst Henn , Stefan Hepper , Klaus Rintdorff, Thomas schack “ Pervasive Computing “ Technology and Architecture of Mobile Internet Applications , Pearson



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Technical Communication
Semester: 2nd
Credits: 00
Pre Requisite: -----

Course Code: MIT591A
Paper Type: Audit
Max Marks: 50
Co-Requisite: -----

Marks Distribution: (Mid Term:15, End Term:25, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *The objective of studying Technical Communication course is that it enhances the employability skills and performance at workplace. It provides discussion on modern media tools for enriching presentation skills for preparing PowerPoint slides. Thus acts as a reference for training programs offered by business houses and Industries.*

COURSE CONTENT:

UNIT I

Technical Communication: Basics of Technical Communication, Barriers to communication and Technology in Communication. Communication in the Workplace: Problem Solving in Workplace Communication, Human factors in the communication failure. Guidelines for ethical communication. Active Listening: Introduction, types of listening, traits of a good listener, Active vs. Passive listening and implications of good listening.

UNIT II

Introduction to Effective Presentation strategies: defining purpose, analysing audience and locale, organizing contents, preparing outline, visual aids, understanding nuances of delivery, kinesics, proxemics, paralinguistic, chromatics. Interviews: introduction, objectives, types of interviews and job interviews, guidelines for surviving a job interview. Group Communication: Introduction, Group discussion, Organizational Group discussions and meeting conferences. Technical Proposals: Definition, purposes, Types, Characteristics, elements of structure, style and Appearance, evaluation.

COURSE OUTCOMES:

At the end of the course, students will demonstrate proficiency by:

- *Understanding the characteristics of technical writing and the importance of purpose, audience, and genre for written communication in technical fields.*
- *Articulating complex engineering ideas appropriate for targeted audiences.*
- *Planning, drafting, revising, editing, and critiquing technical and professional documents through individual and collaborative writing.*
- *Writing effective technical and business documents that are grammatically and stylistically correct.*
- *Preparing and delivering professional technical presentations through applying principles of effective oral communication and slide design.*
- *Applying principles for the visual display of quantitative information.*
- *Researching, analyzing, synthesizing, and applying information to create technical reports.*
- *Recognizing ethical implications of technical communication in professional contexts.*
- *Understanding the contemporary issues in engineering from an environmental, societal, economic, and global perspective.*

Text Books:

1. William Pfeiffer, Padmaja, "Technical Communication A Practical Approach", Pearson Education.

References:

1. Meenakshi Raman & Sangeeta Sharma, "Technical Communication", Oxford University Press



Semester - III

MSc. IT Syllabus – Department of Computer Science, IUST

Semester-III (26 Credit Semester with 31 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT601C	Full-Stack Development	Core	5	3	0	4	07	EC
MIT602C	Artificial Intelligence & Machine Learning	Core	5	3	0	4	07	AEC
MIT603C	Software Engineering	Core	4	3	1	0	04	FC
MIT604C	Design and Analysis of Algorithms	Core	4	3	1	0	04	FC
6 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 3 Discipline Centric Elective Credit Units								
MIT611E	Mobile Communication	DCE	3	2	1	0	03	AEC
MIT612E	Distributed Databases	DCE	3	2	1	0	03	EC
MIT613E	Cryptography and Network Security	DCE	3	2	1	0	03	SEC
MIT614E	System Software Design	DCE	3	2	1	0	03	AEC
MIT615E	Natural Language Processing	DCE	3	2	1	0	03	AEC
Pool B: 3 Discipline Centric Elective Credit Units								
MIT626E	Digital Image Processing	DCE	3	2	0	2	04	AEC
MIT627E	Object Oriented Analysis & Design	DCE	3	2	0	2	04	EC
MIT628E	Data Warehouse and Mining	DCE	3	2	0	2	04	EC
MIT629E	Java Programming	DCE	3	2	0	2	04	EC
02 Open Elective Credit Units (Mandatory)								
**	**	OE	2	2	0	0	02	
Total			26				31	

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Full-Stack Development

Semester: 3rd

Credits: 05

Pre Requisite: MIT501C, MIT526E

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

Course Code: MIT601C

Paper Type: Core

Max Marks:: 125

Co-Requisite: MIT629E

COURSE OBJECTIVES:

- *Understand the fundamentals of ASP.NET Core, MVC architecture, and JavaScript for full-stack web development.*
- *Gain proficiency in creating responsive and dynamic web applications using ASP.NET Core MVC and JavaScript.*
- *Develop skills in backend development, including database connectivity, authentication, and RESTful API implementation.*
- *Learn client-side scripting techniques with JavaScript and jQuery for enhancing user interactivity.*
- *Acquire hands-on experience in deploying and maintaining ASP.NET Core MVC applications on hosting platforms like Azure/ AWS/ GCP.*

COURSE CONTENT

UNIT I

Overview of .NET Framework and .NET Core, Basic syntax and structure of C# programs, Data types, variables, and constants, Conditional statements (if, else, switch), Looping constructs (for, while, do-while, foreach), Exception handling (try, catch, finally).

Classes and objects, Properties, methods, and fields, Constructors and destructors, Inheritance and polymorphism, Basics of Reading from and writing to files.

UNIT II

Introduction to Model-View-Controller (MVC) pattern, Creating models, views, and controllers, Routing and URL mapping, Razor syntax and view engine, Creating and processing forms, Model binding and validation, Tag Helpers and HTML Helpers, Partial views and view components.

Understanding dependency injection (DI) in ASP.NET Core, Configuring and using DI.

Introduction to ASP.NET Core Identity, Implementing authentication and authorization.

UNIT III

Overview of Entity Framework Core (EF Core), Setting up EF Core in an ASP.NET Core project, Creating and configuring the DbContext, Defining and mapping entities, Creating, reading, updating, and deleting (CRUD) operations

LINQ queries and lambda expressions, Database migrations and schema updates, Relationships (one-to-one, one-to-many, many-to-many), Building basic ASP.NET Core application with EF Core.

UNIT IV

Frontend Development with JavaScript and jQuery: Introduction to client-side scripting with JavaScript (ES6+ Concepts), DOM manipulation and event handling, JavaScript functions and closures.

Introduction to jQuery library and its utility in web development, Integrating JavaScript and jQuery with ASP.NET Core MVC applications, Implementing client-side form validation and interactivity.

COURSE OUTCOMES:

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

- *Demonstrate proficiency in setting up development environments for ASP.NET Core MVC and JavaScript projects and to Design & implement MVC architecture for building scalable and maintainable web applications.*
- *Utilize Entity Framework Core for database operations, implement validation and error handling mechanisms, also implement authentication and authorization features using ASP.NET Identity for secure user management.*

School of Engineering & Technology (SoE&T),

Islamic University of Science & Technology (IUST),

1-University Avenue, Awantipora, Pulwama, Pin -192122, Jammu and Kashmir.

Page - 65

MSc. IT Syllabus – Department of Computer Science, IUST

- *Develop interactive user interfaces using JavaScript and jQuery for dynamic content manipulation and event handling.*
- *Integrate frontend and backend components seamlessly through AJAX requests and API consumption.*
- *Construct single-page applications (SPAs) using ASP.NET Core MVC and JavaScript frameworks for enhanced user experience.*
- *Employ testing and debugging techniques to ensure the functionality and reliability of full-stack applications.*
- *Deploy ASP.NET Core MVC applications to cloud hosting platforms and manage application lifecycles effectively.*

Text Books:

1. Freeman, Adam. "Pro ASP.NET Core MVC 2." Apress, 2017.
2. Duckett, Jon. "JavaScript and JQuery: Interactive Front-End Web Development." Wiley, 2014.

References:

1. Lock, Andrew. "ASP.NET Core in Action." Manning Publications, 2018.
2. Esposito, Dino. "Programming ASP.NET Core." Microsoft Press, 2018.
3. Flanagan, David. "JavaScript: The Definitive Guide." O'Reilly Media, 2020.
4. Freeman, Adam. "Pro jQuery in ASP.NET Core MVC." Apress, 2018.
5. Schwartz, Carl. "A Practical Guide to ASP.NET Web API." Apress, 2013.
6. Wilson, James, et al. "Professional ASP.NET MVC 5." Wrox, 2014.

List of Practicals:

LAB JOURNAL_ MIT601C	
Week No	TITLE
1 to 4	<ul style="list-style-type: none">• Set up development environment with Visual Studio and create a basic ASP.NET Core MVC project.• Implement models, views, and controllers for a simple CRUD (Create, Read, Update, Delete) application.• Configure routing and request handling in ASP.NET Core MVC.• Enhance the CRUD application by adding form validation and error handling mechanisms.
5 to 8	<ul style="list-style-type: none">• Integrate Entity Framework Core for database connectivity and implement data models.• Implement authentication and authorization using ASP.NET Identity.• Extend the application to include RESTful APIs for data manipulation.• Deploy the application to a local development server or cloud platform like Azure.
9 to 12	<ul style="list-style-type: none">• Integrate JavaScript for client-side interactivity and basic DOM manipulation.•• Implement form validation using JavaScript.• Explore advanced JavaScript concepts like asynchronous programming and closures.• Integrate jQuery library into the ASP.NET Core MVC project for enhanced DOM manipulation and event handling.
13 to 16	<ul style="list-style-type: none">• Implement AJAX requests to retrieve and update data asynchronously from the server.• Build a single-page application (SPA) using JavaScript frameworks like React, Angular, or Vue.js with ASP.NET Core MVC backend.• Final project: Students develop a full-stack web application that incorporates all concepts learned throughout the course, including backend connectivity, client-side interactivity, and real-time communication.

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Artificial Intelligence & Machine Learning	Course Code:	MIT602C
Semester:	3 rd	Paper Type:	Core
Credits:	05	Max Marks::	125
Pre Requisite:	MIT552C, MIT579E	Co-Requisite:	MIT604C
Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- To explain Artificial Intelligence and different machine learning procedures which include Supervised, Unsupervised, semi supervised and reinforcement learning etc.
- To explain rational AI agents and familiar machine learning Techniques like Support Vector Machines and its nonlinear variant.
- To explain few distribution like Gaussian, Dirichlet etc. and its subsequent use in modelling data for learning.
- To explain common applications of artificial intelligence and machine learning with a specific focus on latest developments in the said area.

COURSE CONTENT:

UNIT I

Artificial Intelligence and its Agents: Introduction to Artificial Intelligence, Applications of Artificial Intelligence with respect to various fields, Intelligent Agents- Agents and Environments, Structure of Agents, Concept of Rationality.

Introduction to Proposition Logic and FOL: Introduction to Proposition Logic, First Order Logic(FOL), Rules of Inference.

Search Algorithms: UnInformed Search Algorithms(BFS, DFS, Uniform Cost Search etc), Informed Search (Heuristic Search, Greedy Best first Search, A* Search)

UNIT II

Foundations: Definition of learning systems, learning types (supervised, unsupervised, reinforcement).Goals and applications of machine learning in various domains.

Machine learning process overview: data acquisition, cleaning, pre-processing, feature engineering, model selection, training, evaluation, deployment.

Concept Learning: Hypothesis space and inductive bias. General-to-specific ordering of hypotheses and greedy search algorithms. Importance of simplicity and overfitting avoidance. Introduction to model evaluation metrics (accuracy, precision, recall, F1-score).

UNIT III

Supervised Learning/Classification models:-

Decision Trees: ID3 algorithm, entropy-based attribute selection. Overfitting, pruning techniques, and their impact. Ensemble methods: Bagging, Boosting.

Linear Regression: Linear, multiple.

Bayesian Learning: Probability theory basics (conditional probability, Bayes' rule).Naive Bayes classifier and parameter smoothing. Logistic regression for classification.

Support Vector Machines (SVMs): Maximum margin hyperplane concept and kernel methods. Applications of SVMs in different domains.

Artificial Neural Networks (ANNs): Biological inspiration, perceptron's and limitations. Multilayer networks, backpropagation algorithm. Overfitting prevention techniques (regularization, dropout, cross-validation).

MSc. IT Syllabus – Department of Computer Science, IUST

UNIT IV

Unsupervised Learning/ Clustering Models: - Hierarchical clustering and k-means clustering. Applications of clustering in data exploration and segmentation. k-Nearest Neighbor (k-NN) algorithm and its variations. Applications of k-NN in recommendation systems and anomaly detection.

Deep Learning (Introduction): Brief overview of deep learning architectures (Convolutional neural networks, recurrent neural networks). Applications of deep learning in computer vision, natural language processing, and other fields.

COURSE OUTCOMES:

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

- Apply a variety of learning algorithms to data.
- Perform evaluation of learning algorithms and model selection?
- Apply machine learning algorithms to solve problems of moderate complexity

Text Books:

1. Michalski, Ryszard S., Jaime G. Carbonell, and Tom M. Mitchell, eds. Machinelearning: An artificial intelligence approach. Springer Science & Business Media, 2013.

References:

1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
2. Introduction to Machine Learning Edition 2, by Ethem Alpaydin

List of Practicals:

LAB JOURNAL_ MIT602C	
S. No.	TITLE
1.	NumPy i. Write a program in python to create a Numpy array ii. Write a program in python to count the number of non-zero values in the array iii. Write a program in python to count the number of elements along a given axis in Numpy array iv. Write a program in python to reverse a numpy array
2.	Pandas/Matplotlib i. Write a program in python to make a Pandas DataFrame with two-dimensional list ii. Write a program in python for ranking Rows of Pandas DataFrame iii. Write a program in python for Sorting rows in pandas DataFrame iv. Write a program in python to increase the size of scatter points in Matplotlib ?
3.	Machine Learning i. Write a python script to implement Linear Regression using sklearn ii. Write a python script to implement Logistic Regression using Tensorflow iii. Write a python script to implement Naive Bayes using SKlearn iv. Write a python script to implement Support Vector Machine Algorithm using SKLearn v. Write a python script to implement K-means Algorithm using SKLearn
4.	Deep Learning i. Write a python script to implement ANN from scratch ii. Write a python script to implement artificial neural network with Keras iii. Write a python script to implement Dog-Cat classifier using keras API iv. Write a python script to Implement Dog-Cat Classifier using PyTorch

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Software Engineering	Course Code:	MIT603C
Semester:	3 rd	Paper Type:	Core
Credits:	04	Max Marks:	100
Pre Requisite:	MIT501C, MIT552C	Co-Requisite:	MIT601C, MIT628E, MIT627E
Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To familiarize students with the fundamentals of software engineering.*
- *To introduce students to the basic concepts and principles of structured software engineering.*
- *To make students understand the software management issues.*
- *To introduce the students to the basic concepts of OOSE.*
- *To study advanced concepts in software engineering like Reverse engineering, Reengineering, etc.*

COURSE CONTENT:

UNIT I

Software Engineering: Definition & Evolution, its Role & Impact in Computer Science. Software Process, Characteristics of a SW Process, CMMI, TSP & PSP, Software Product, Characteristics of a Good Software Product, Software Process Models, Comparative Study & Applications. Basic concepts of Agile Process. Software Requirements Analysis (SRA): Requirements - Types, Steps Involved in SRA. SW Requirements Specification (SRS): Need & Characteristics for an SRS, Components of an SRS, Prototype for a Good SRS. Structured Analysis: DFD'S, Control Flow Diagrams, Data Dictionary, State Transition Diagrams, & Entity - Relationship Diagrams. Case Study: Developing a Complete SRS.

UNIT II

Software Design: Concepts & Principles, Design Considerations & Good Design. Characterization of Effective Modular Design (Functional Independence, Cohesion, Coupling). Design: Architectural Design, Procedural Design, Interface Design, & Data Design. SW Architecture Styles: (Dataflow, Call & Return Architectures, Independent Process Architectures, Virtual Machine Architectures). Concept of Verification & Validation. Goals of SW Testing, Testing Principles.

UNIT III

Approaches to the Design of Test Cases: Black Box & White Box Testing, Techniques used by these Approaches: Basis Path & Loop Testing, Graph Based Testing, Equivalence Partitioning, Cyclomatic Complexity, Documentation of Test Cases, Phases in Testing Activity: Unit,

Integration, Validation & System Tests. Software Project Management, Phases of Management, Project Planning & Control, Scheduling, Organization & Team Structures, Project Estimation Techniques – KLOC, FP & COCOMO, Risk Analysis & Management, , "Risk Exposure", Software Quality Assurance, Software Configuration Management.

UNIT IV

Technical Metrics for Software. Object Oriented Software Engineering: Object Oriented Paradigm, Concepts - Classes & Objects, Inheritance, Abstraction, Polymorphism, etc. OOA & OOD. Design Methodology – Dynamic Modeling, Functional Modeling. Advanced Concepts: Software Reuse, Re-engineering, Reverse Engineering, Restructuring, Client/Server Software Engineering, Computer Aided Software Engineering, Advances & Future Scope in Software Engineering.

COURSE OUTCOMES:

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

- *Describe software engineering layered technology and process framework.*

MSc. IT Syllabus – Department of Computer Science, IUST

- *Understand theories, models, and techniques that provide a basis for the software development life cycle.*
- *Understand software testing approaches including verification and validation, static analysis, reviews, inspections, and audits.*
- *Understand the role of project management including planning, scheduling, risk management, etc.*
- *Work as an individual and/or in team to develop and deliver quality software.*

Text Books:

1. Pressman, Roger, “Software Engineering- A Practitioners Approach”, McGraw Hill

References:

1. Gheezi, Jazayeri Et Al, “Fundamentals Of Software Engineering”, PHI
2. Ian Sommerville, “Software Engineering”, Pearson Education
3. PankajJalote, “An Integrated Approach To Software Engineering”, Narosa
4. Peters & Pedrycz, “Software Engineering an Engineering Approach”, Wiley



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Design and Analysis of Algorithms	Course Code:	MIT604C
Semester:	3 rd	Paper Type:	Core
Credits:	04	Max Marks:	100
Pre Requisite:	MIT501C, MIT551C, MIT552C	Co-Requisite:	MIT602C
Marks Distribution:	(Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)		

COURSE OBJECTIVE:

- *The objective of the course is to teach techniques for effective problem solving in computing. The use of different paradigms of problem solving will be used to illustrate clever and efficient ways to solve a given problem. In each case, emphasis will be on rigorously proving correctness of the algorithm. In addition, the analysis of the algorithm will be used to show the efficiency of the algorithm over the naive techniques.*

COURSE CONTENT:

UNIT I

Algorithms: Introduction, Importance, Performance Analysis, Designing Algorithms, Growth of Functions, Asymptotic Notations, Solving Recurrences. Review of Data Structures: Stacks, Queues, Trees, Graphs & Hashing.

UNIT II

Divide & Conquer: General Method, Binary Search, MinMax Problem, Merge Sort, Quick Sort, Strassen's Matrix Multiplication. Greedy Method: General Method, Knapsack Problem, Job Sequencing with Deadlines, Minimum Cost Spanning Trees, Optimal Storage on Tapes, Optimal Merge Pattern, Single Source Shortest Paths.

UNIT III

Dynamic Programming: General Method, Multistage Graphs, All Pair Shortest Paths, Traveling Salesperson Problem. Backtracking: General Method, N-Queen Problem, Sum of Subsets Problem, Graph Coloring, 0-1 Knapsack Problem.

UNIT IV

Branch & Bound: General Method, Least Cost Branch & Bound, 8-Queen Problem. Lower Bound Theory: Comparison Trees, Lower Bounds through Reductions, P & NP Problems, NP Hard & NP Complete Problems, Handling of NP Hard Problems. Parallel Algorithms: Basic Concept, Architectures, Effect of Parallelism.

COURSE OUTCOMES:

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

- *Design and analyze programming problem statements.*
- *Understand the necessary mathematical abstraction to solve problems.*
- *Come up with analysis of efficiency and proofs of correctness.*
- *Comprehend and select algorithm design approaches in a problem specific manner.*

Text Books:

1. Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein, MIT Press, Third Edition, 2009.

References:

1. Algorithms, by Dasgupta, Papadimitrou and Vazirani, McGraw-Hill Education, 2006.
2. Computer Algorithms, by Horowitz, Sahni, and Rajasekaran, Silicon Press, 2007.
3. Algorithm Design, by Kleinberg and Tardos, Pearson, 2005.
4. Algorithm Design, by Goodrich and Tamassia, Wiley, 2001.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Mobile Communication
Semester: 3rd
Credits: 03
Pre Requisite: MIT553C

Course Code: MIT611E
Paper Type: DCE
Max Marks: 75
Co-Requisite: MIT601C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *The objective of this course is to give an introduction to the fundamentals of the wireless communications systems, the wireless network architectures, protocols, and applications.*
- *To provide an overview of Wireless Communication networks area and its applications in communication engineering.*
- *To enable the student to synthesis and analyze wireless and mobile cellular communication systems over a stochastic fading channel*
- *To provide the student with an understanding of advanced multiple access techniques.*
- *To provide the student with an understanding of diversity reception techniques*
- *To give the student an understanding digital cellular systems (cdma2000, W-CDMA, LTE and candidate 5G waveforms).*

COURSE CONTENT:

UNIT I

Wireless Communication Systems: Basic Concepts, Wireless Networks, Wireless vs Wired networks, Classification of Wireless Networks. Wireless Transmission Concepts: Antennas, Types of antennas, Signal propagation, Multipath propagation. Types of Wireless Telephones (Cordless, Fixed Wireless (WLL), Wireless With Limited Mobility (WLL-M), Fully Mobile Wireless Phones). Introduction to various Generations of Mobile Phone Technologies. Wireline/Wireless Portion of Mobile Communication Network.

UNIT II

Electromagnetic spectrum, its use & allocation to well-known bands. Multiple Access Techniques: FDMA, TDMA & CDMA. Basic concepts of Spread Spectrum(SS) technique; Direct Sequence SS versus Frequency Hopping SS. Concept of cells, sectorization, coverage area, frequency reuse, cellular networks, microcells & handoffs. Simplified Implementation of IS-95 CDMA Using Chip Sequences, Concept of (Walsh) Code, Concept of CDMA (PCS and Cellular) Channel, Forward & Reverse CDMA Channel for a Cell/Sector, Comparison of Cellular/PCS CDMA Networks, Frequencies & Cell-Sizes.

UNIT III

Cellular Mobile Systems: Introduction & Overview of GSM/CDMA. Channel: Types, Forward & Reverse CDMA channel for a cell/sector (Pilot, Sync, Paging, Forward Traffic Channels, Access, Reverse TC). Concept of Voice Coding. Components of Mobile Network Architecture: MS, BTS, BSC, MSC, & their basic Functions. Types of Handoffs in GSM. Use of HLR & VLR in Mobile

Networks. Concept of Mobile IP & WAP. Wireless LAN Technology: Overview of Wifi & the IEEE 802.11. Concept of Bluetooth & IEEE 802.15.

COURSE OUTCOMES:

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

- *Demonstrate advanced knowledge of networking and wireless networking and understand various types of wireless networks, standards, operations and use cases.*
- *Demonstrate knowledge of protocols used in wireless networks and learn simulating wireless networks.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. T. Rappaport, “Wireless Communications, Principles and Practice”, Pearson and Dornan, “The Essential Guide to Wireless Communications”, Pearson
2. Jochen Schiller, “Mobile Communications”, Pearson and Andrew Tanenbaum, “Computer Networks”, PHI

References:

1. William Stallings, “Wireless Communications & Networks”, Pearson
2. K.Pahlavan, P.Krishnamurthy, “Principles of Wireless Networks”, Pearson Education.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Distributed Databases	Course Code:	MIT612E
Semester:	3 rd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT502C	Co-Requisite:	MIT604C, MIT628E
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies and showing the need for distributed database technology to tackle deficiencies of the centralized database systems.*
- *To expose active and emerging research issues in distributed database systems and application development.*

COURSE CONTENT:

UNIT I

Features of Distributed versus Centralized Databases – Why Distributed Databases – Distributed Database Management Systems (DDBMSs)- Review of Databases – Review of Computer Networks-Levels of Distribution Transparency- Reference Architecture for Distributed Databases – Types of Data Fragmentation – Distribution Transparency for read-only Applications – Distribution transparency for Update Applications – Distributed Database Access Primitives – Integrity Constraints in Distributed Databases - A Framework for Distributed Database Design – The Design of Database Fragmentation – The Allocation of Fragments.

UNIT II

Foundations of Distributed Concurrency Control – Distributed Deadlocks – Concurrency Control Based on Timestamps – Optimistic Methods for Distributed Concurrency Control - Reliability – Basic Concepts Non-blocking Commitment Protocols – Reliability and Concurrency Control – Determining a Consistent View of the Network – Detection and Resolution of Inconsistency – Checkpoints and Cold Restart - Distributed Database Administration – Catalog Management in Distributed Databases – Authorization and Protection.

UNIT III

Distributed object database management systems – Fundamental object concepts and Models – Object – Abstract Data Types – Composition (Aggregation) – Class – Collection – Subtyping and Inheritance. – Object Distribution Design – Horizontal Class Partitioning – Vertical Class Partitioning – Path Partitioning – Class Partitioning Algorithms – Allocation – Replication – Alternative Client / Server Architectures – Cache Consistency – Object Identifier Management – Pointer Switching Object Migration – Distributed Object Storage – Object Query Processor Architectures – Query Processing Issues – Query Execution – Correctness Criteria – Transaction Models and Object Structures – Transactions Management in Object DBMSs – Transactions as Objects – Conclusion – Bibliographic Notes – Exercises.

COURSE OUTCOMES:

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

- *Explain the techniques used for data fragmentation, replication, and allocation during the distributed database design process.*
- *Evaluate simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer.*
- *Describe distributed concurrency control based on the distinguished copy techniques and the voting methods*
- *Apply theory to practice by building and delivering a distributed database query engine, subject to remote Web service calls.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. M.Tamer Ozsu, Patrick Valduriez, Distributed database systems, 2nd Edition, Prentice Hall of India, New Delhi.
2. Distributed Databases: Principles and Systems by Stefano Ceri, McGrawHill

References:

1. Distributed Database Systems by Chhanda Ray, Pearson
2. Distributed Database Management Systems: A Practical Approach by Frank S. Haug and Saeed K. Rahimi, Wiley
3. Distributed Database Systems by David A. Bell, Jane B. Grimson, and Jane Grimson, Addison-Wesley



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Cryptography & Networks Security

Semester: 3rd

Credits: 03

Pre Requisite: MIT553C

Course Code: MIT613E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT611E, MIT604C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.*
- *To Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.*
- *Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.*
- *Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.*

COURSE CONTENT:

UNIT I

Introduction: Security trends, the OSI Security Architecture, Security attacks, Security mechanisms, Security services, Model for Network Security. Encryption: Symmetric & Asymmetric Encryption, Symmetric Cipher model, Substitution techniques (Mono-alphabetic & Poly-alphabetic) & Transposition Technique, Rotor Machines, Block and Stream ciphers, Modern Round Ciphers: DES, AES, Introduction to Steganography. Encryption vs Steganography.

UNIT II

Introduction to Number Theory: Prime Numbers, Fermat's & Euler's theorem & discrete logarithms. Public Key Cryptography & RSA. Key management: Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic & Elliptic Curve Cryptography. Message Authentication & Hash functions. Digital Signatures & Authentication protocols. Network Security Applications: IP Security

UNIT III

Mechanics of Routing Protocols: Routing, Static and Dynamic Routing, Distant vector routing & link state routing, Internet working with Dissimilar Protocols, Protocol design Consideration: Simplicity, flexibility, Optimality, Overhead & Scaling, Operation above Capacity, forward compatibility. Migration: Routing Algorithms & addressing parameters, making multi-protocol operation possible. Network Security: Security Features in Wireless Adhoc & Wireless Sensor Networks.

COURSE OUTCOMES:

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

- *Evaluate the security treats in modern computer era*
- *Define and identify firewall and network filtering*
- *List and recognize various VPN*
- *Identify different technique of sandboxing*
- *Distinguish various ethical hacking and testing procedures*

Text Books:

1. William Stalling, "Cryptography & Network Security", Pearson Education

References:

School of Engineering & Technology (SoE&T),

Islamic University of Science & Technology (IUST),

MSc. IT Syllabus – Department of Computer Science, IUST

1. Radia Perlman, “Interconnections: Bridges, Routers switches & Internet-working protocols Pearson education
2. Mark Sportack, “IP Routing Fundamentals”, Pearson Education
3. Gerard J. Holzmann, “Design & Validation Computer Protocols”, Prentice Hall



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	System Software Design	Course Code:	MIT614E
Semester:	3 rd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT501C, MIT551C, MIT563E	Co-Requisite:	MIT627E
Marks Distribution:	(Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)		

COURSE OBJECTIVE:

- *The objective of this course is to teach procedures for the design of software systems and to provide a basis for judgement in the design of system software. The focus is to discuss the design and implementation of the major system software components with the consideration of underlining the pertinent design issues. The design issues are likely to be discussed in the context of modern computer systems that include the different languages and advanced operating systems.*

COURSE CONTENT:

UNIT I

Overview, Introduction to Machine Structure, Evolution of the System Programming Components: Assemblers, Loaders, Macros, Linkers, Compilers.

Evolution of Operating System & User View: Functions, Batch Control Language & Facilities.

Machine Structure 360-370: Memory, Registers, Data, Instructions & Special Features. Machine Language & Assembly Language.

UNIT II

Assemblers: General Design Procedure: Problem Statement, Data Structures, Format of Databases, Algorithms, Modularity Lookup. Table Processing: Review of Searching & Sorting Techniques. Macros: Macro Instructions, Features, Conditional Macro Instructions, Macro Calls within Macros, Single & Two Pass Algorithm, Implementation of Macro Calls within Macros. Implementation within an Assembler

UNIT III

Loaders: Schemes: Compile & Go Loaders, General Loader Scheme, Absolute Loaders, Subroutine Linkages, Relocating Loaders. Other Loader Schemes & Binders: Linking Loaders, Overlays & Dynamic Binders. Design of an Absolute Loader, Design of a Direct Linking Loader. Programming Languages & Formal Systems: Significance of High Level Languages, Features. Data Types & Data Structures, Storage Allocation & Scope of Names, Accessing Flexibility, Functional Modularity, Use of Formal Systems, Formal Specification, Formal Grammars, Hierarchy of Languages, Backus Normal Form – BNF, Canonic Systems & Formal Systems.

COURSE OUTCOMES:

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

- *Basic Understanding of Machine Structure and its Major System Software Components*
- *Comprehend Assembler Design Issues and Procedures*
- *Understand Macros and Macro Challenges*
- *Understand various Loader Schemes with the consideration of underlining the pertinent design issues*
- *Understand Compilers in the context of different languages and advanced operating systems.*

Text Books:

1. John J. Donovan, "Systems Programming", Tata McGraw-Hill

References:

1. Barron.D.W. "Assemblers & Loaders", Mc Donald & Javes
2. Ullman.J.D."Fundamentals of Programming System" Addison & Wesley.
3. M.Dhamdhere."System Programming & Operating Systems"



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Natural Language Processing	Course Code:	MIT615E
Semester:	3 rd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT552C, MIT579E	Co-Requisite:	MIT602C, MIT604C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To understand natural language processing and to learn how to apply basic algorithms in this field.*
- *To get acquainted with the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora.*
- *To conceive basics of knowledge representation, inference, and relations to the artificial intelligence.*

COURSE CONTENT:

UNIT I

The study of Language, challenges of NLP, The Different Levels of Language Analysis, Representations & Understanding, Linguistic background, Grammars & Parsing, Features & Augmented Grammars, Grammars for Natural Language, Towards Efficient Parsing, Ambiguity Resolution. Applications in Information Retrieval, design features of information retrieval system, Language Modelling: Grammar based model & statistical model.

UNIT II

Word Level & Syntactic Analysis: Spelling error detection & correction, parts of speech tagging Morphological parsing, Probabilistic parsing .Semantic Interpretation: Semantics & Logical Form, Linking Syntax & Semantics, Strategies for Semantic Interpretation. Pragmatics: Reference Resolution, Syntactic & Semantic Coherence, Text Coherence, An Inference based resolution algorithm.

UNIT III

Natural language Generation: Introduction to language generation, architecture for generation, surface realization, systemic grammar, functional unification grammar, discourse planning Machine Translation: problems in machine translation, Language Similarities & Differences, transfer metaphor, syntactic transformations, lexical transfer, idea of Interlingua, direct translation, using statistical Techniques.

COURSE OUTCOMES:

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

- *The students will get acquainted with natural language processing and learn how to apply basic algorithms in this field.*
- *They will understand the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language processing.*
- *They will also grasp basics of knowledge representation, inference, and relations to the artificial intelligence.*

Text Books:

1. James Allen, "Natural Language Understanding", Pearson Publication, ISBN: 978-81-317-08958 2nd Edition

References:

1. D. Jurafsky, J. H. Martin, "Speech & Language Processing", Pearson Education, 2002.
2. Speech & Language Processing, by Jurafsky, D. & Martin,
3. J.H.Tanveer Siddiqui, US Tiwary, Natural Language Processing & Information Retrieval

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Digital Image Processing	Course Code:	MIT626E
Semester:	3 rd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT552C, MIT579E	Co-Requisite:	MIT604C, MIT614E
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To study image fundamental and mathematical transforms necessary for image processing.*
- *To study fundamental technologies for digital image, compression, analysis, and processing*
- *To Gain understanding of algorithm, analytical tools, and practical implementations of various digital image applications*

COURSE CONTENT:

UNIT I

Introduction & Digital Image Fundamentals: The origins of Digital Image Processing, Examples of Fields that Use Digital Image Processing, Fundamentals Steps in Image Processing, Elements of Digital Image Processing Systems, Image Representation, Image Sampling & Quantization, Some basic relationships like Neighbors, Connectivity, Distance Measures between pixels, Linear & Non Linear Operations. Image Enhancement in the Spatial Domain: Some basic Gray Level transformations, Histogram Processing, Enhancement Using Arithmetic & Logic operations, Basics of Spatial Filters, Smoothing & Sharpening Spatial Filters, Combining Spatial Enhancement Methods.

UNIT II

Image Enhancement in the Frequency Domain: Introduction to Fourier Transform & the frequency Domain, Smoothing & Sharpening Frequency Domain Filters, Homomorphism Filtering. Image Restoration: A model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Dedradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Constrained Least Square Filtering, Geometric Mean Filter, Geometric Transformations.

UNIT III

Image Compression: Coding, Interpixel & Psychovisual Redundancy, Image Compression models, Elements of Information Theory, Error free comparison, Lossy compression, Image compression standards. Image Segmentation: Detection of Discontinuities, Edge linking & boundary detection, Thresholding, Region Oriented Segmentation, Motion based segmentation.

COURSE OUTCOMES:

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

- *Understand Basics of Image formation and transformation using sampling and quantization.*
- *Perform and apply compression and coding techniques used for image data.*

Text Books:

1. Rafael C. Gonzalez & Richard E. Woods, "Digital Image Processing", Pearson Education

References:

1. A.K. Jain, "Fundamental of Digital Image Processing", PHI
2. Bernd Jahne, "Digital Image Processing", Springer
3. William K Pratt, "Digital Image Processing: Piks Inside", John Wiley & Sons

MSc. IT Syllabus – Department of Computer Science, IUST

List of Practicals:

LAB JOURNAL_ MIT626E	
S. No.	TITLE
1.	Image Representation & Sampling Convert an image into its digital representation (e.g., grayscale or RGB). Demonstrate the effect of different sampling rates on image quality.
2.	Image Quantization & Basic Operations Quantize the image using different bit-depths. Perform basic operations like addition, subtraction, and multiplication on images.
3.	Neighbors, Connectivity, & Distance Measures Implement algorithms to find neighboring pixels. Calculate distance measures like Euclidean distance between pixels.
4.	Linear & Non-Linear Operations Implement linear operations like contrast stretching and gamma correction. Implement non-linear operations like thresholding and histogram equalization.
5.	Spatial Filters Implement basic spatial filters like mean and median filters. Demonstrate the effects of different filter sizes on image quality.
6.	Combination of Spatial Enhancement Methods Combine spatial enhancement methods (e.g., filtering followed by histogram equalization) and compare results.
7.	Fourier Transform & Frequency Domain Filters Implement the 2D Fourier transform. Apply frequency domain filters like low-pass and high-pass filters.
8.	Image Restoration Techniques Implement inverse filtering for deconvolution. Implement Wiener filtering for noise reduction.
9.	Periodic Noise Reduction Implement frequency domain filtering to reduce periodic noise. Demonstrate the effect of different filter parameters on noise reduction.
10.	Geometric Transformations Implement basic geometric transformations like rotation and scaling. Apply geometric transformations to images and analyze the results.
11.	Image Coding & Compression Implement basic coding and compression techniques like Huffman coding. Analyze the compression ratio and image quality.
12.	Psychovisual Redundancy Analyze the psychovisual redundancy of different image regions. Demonstrate the effect of psychovisual redundancy on compression efficiency.
13.	Image Segmentation Techniques Implement edge detection algorithms like Sobel and Canny. Implement region-oriented segmentation using clustering algorithms.
14.	Motion-Based Segmentation Implement motion-based segmentation using background subtraction techniques. Analyze the effect of different motion models on segmentation accuracy.

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Object Oriented Analysis & Design	Course Code:	MIT627E
Semester:	3 rd	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT501C, MIT552C	Co-Requisite:	MIT601C, MIT603C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVE:

- *To understand the definition of problems, requirements specification and the relationship between the specification and the real-world*
- *To Understand systems specification, modelling and a superior functional design and a variety of systems development strategies*
- *To understand the process of transforming Use Cases into Object Oriented software Realizations through OO Analysis and OO Design.*
- *To Understand the essential and fundamental aspects of object-oriented analysis and design, in terms of “how to use” it for the Purpose of specifying and developing software.*

COURSE CONTENT:

UNIT I

Introduction to OOAD, Structures Design & Unified Process, Class & objects, effect of inheritance on polymorphism & variable declarations, concepts that define object orientation.

UNIT II

Requirements: Developing requirements, reviewing requirements, managing requirements, Difficulties & risks in domain & requirement analysis, requirement documents

UML diagrams, Use Case Diagrams, System sequence diagrams, Relationship between sequence diagrams and use cases, Class Diagrams, Interaction Diagrams, State Diagrams, Difficulties & risks in creating class diagram.

UNIT III

Activity Diagrams, Package, component and Deployment Diagrams: Case study – the Next Gen POS system. Architecting & Designing Software: The process of design, design principles, User interface design. GRASP: Designing objects with responsibilities, Creator, Information expert, Low Coupling, High Cohesion, Controller, Design patterns – Singleton, observer, adapter, Façade, Proxy with examples.

COURSE OUTCOMES:

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

- *Demonstrate the importance of modelling in the software development life cycle.*
- *Become familiar with the Unified modelling Language.*
- *Understand the object-oriented approach to analysing and designing systems and software solutions. Employ the Unified modelling Language notations to create effective and efficient system designs.*
- *Understand the difference between writing programs for the software and doing analysis and design.*
- *Problem formulation and decomposition (analysis) and solution building (design) will be covered.*

Text Books:

1. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition By Craig Larman
2. Analysis Patterns Reusable Object Models, Martin Fowler
3. Design Patterns: Elements of Reusable Object-Oriented Software Book by Erich Gamma, John Vlissides, Ralph Johnson, and Richard Helm

MSc. IT Syllabus – Department of Computer Science, IUST

References:

1. Object Oriented Analysis & Design With Applications (Grady Booch ,Robert A.Maksimchuk)
2. Head First-Objected Oriented Analysis & Design
3. Object-Oriented Analysis & Design Using UML (D Jeya Mala, Mc Graw Hill Education)

List of Practicals:

LAB JOURNAL_ MIT630E	
S. No.	TITLE
1.	Write a Program in C++/JAVA to demonstrate the concept of Inheritance.
2.	Requirement Documents: Case study of NextGen POS System/ University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System.
3.	Use Cases: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
4.	Use Case Diagrams: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
5.	Domain Class Diagrams: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
6.	System Sequence Diagrams from Use Cases: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
7.	Object Diagrams: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
8.	Design Class Diagram: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
9.	Activity Diagram: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
10.	State Diagrams: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System
11.	Package and Deployment Diagrams: Case study of NextGen POS System/University Transport System/Blood Donation System/ Estate management System/Organ Donation Management System/Online Product Review System/ Examination Automation System

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Data Warehouse and Mining
Semester: 3rd
Credits: 03
Pre Requisite: MIT502C

Course Code: MIT628E
Paper Type: DCE
Max Marks: 75
Co-Requisite: MIT612E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To provide students with an understanding of the fundamentals of data warehouse, its basic components and the role of data warehouse in decision support.*
- *To provide knowledge of the important steps and techniques to be considered during data warehouse development, future trends and usage of data warehouse.*
- *To introduce the concept of dimensional modelling technique for designing a data warehouse.*
- *To provide knowledge of data integration and the extraction, transformation and load (ETL) processes.*
- *To introduce students to the methods and tools for accessing and analyzing the warehouse data.*

COURSE CONTENT:

UNIT I

Overview of decision support systems: Organizational need for strategic information, Failures of past decision-support systems, operational versus decision-support systems, data warehousing – the only viable solution, data warehouse defined. Data warehouse – The building Blocks: Defining Features, data warehouses & data marts, overview of the components, metadata in the data warehouse

Defining the business requirements: Dimensional analysis, information packages – a new concept, requirements gathering methods, requirements definition: scope & content

UNIT II

Principles of dimensional modeling: Objectives, From Requirements to data design, the STAR schema, STAR Schema Keys, Advantages of the STAR Schema

Dimensional Modeling: Updates to the Dimension tables, miscellaneous dimensions, the snowflake schema, aggregate fact tables.

UNIT III

Data Extraction, Transformation and Loading: Objectives, ETL Overview, ETL Requirements and Steps, Data Extraction- Source Identification, Data Extraction Techniques, Evaluation of Techniques, Data Transformation-Major Transformation Types, Data Integration and

Consolidation, Transformation for Dimension Attributes, Transformation Implementation, Data Loading, ETL Tool Options, Other Integration Approaches.

COURSE OUTCOMES:

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

- *Understand the concept related to data warehouse architecture*
- *Compare Online Analytical Processing (OLAP) and Online Analytical Transaction Processing(OLTP) tools*
- *Design a data mart or data warehouse for any organization*
- *Asses raw input data and preprocess it to provide suitable input for range of data mining algorithms*
- *Extract association rules and classification model*
- *Identify the similar objects using clustering techniques*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. Paul Raj Poonia, "Fundamentals of Data Warehousing", John Wiley & Sons

References:

1. Sam Anahony, "Data Warehousing in the real world: A practical guide for building decision support systems", John Wiley
2. Alex Berson, Stephen J. Smith, "Data Warehousing, Data Mining & OLAP", Tata McGraw Hill W. H. 5.
3. Inmon, "Building the operational data store", John Wiley
4. Kamber & Han, "Data Mining Concepts & Techniques", Hartcourt India P. Ltd.

List of Practicals:

LAB JOURNAL_ MIT628E	
S. No.	TITLE
1. K	Based on a given case study: a. Define business requirements b. Perform Dimension Analysis c. Design information packages d. Do requirement gathering
2.	Based on a given case study: a. Design of fact & dimension tables b. Generating graphs for star schema c. Generation of snowflake schema
3.	Identify source tables and populate sample data.
4.	Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking-Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc.).
5.	Write ETL scripts and implement using data warehouse tools
6.	Perform Various OLAP operations such slice, dice, roll up, drill up and pivot.
7.	Explore visualization features of the tool for analysis like identifying trends etc.
8.	Explore WEKA Data Mining/Machine Learning Toolkit

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Java Programming
Semester: 3rd
Credits: 03
Pre Requisite: MIT501C, MIT552C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT629E
Paper Type: DCE
Max Marks: 75
Co-Requisite: MIT601E, MIT627E

COURSE OBJECTIVES:

- To cover OOPS concepts like inheritance, polymorphism, overriding etc.
- To introduce Lambda expressions, inner classes and interfaces.
- To learn about threads, assertions, logging and exception handling.
- To know about collection framework and introduction of spring framework.

COURSE CONTENT:

UNIT I

Fundamental Programming structures in Java: A simple java program, comments, data type, variables, operators, string, input& output, control flow, big numbers, arrays. Objects & Classes: introduction to Object Oriented programming, using predefined classes, defining your own classes, static fields & methods, method parameters, object construction, packages, the class path, documentation comments.

UNIT II

Inheritance Classes superclasses, subclasses, object: the cosmic superclass, generic array list, object wrapping & autoboxing, method with a variable number of parameters, enumeration class, reflection, design hints for inheritance. Interfaces lambda expressions & inner classes. Interfaces, examples of interfaces, lambda expressions, inner classes, proxies. Exceptions, assertion & logging dealing with errors, catching exception, using assertion, logging.

UNIT III

Generic Programming why generic programming, defining a simple generic class, generic methods, bound for type variables, generic code & virtual machine restriction & limitation, inheritance rules for generic types, wildcard types, reflection & generics. Collections Java collection framework, concrete frameworks, Maps, Views & wrappers, Algorithms, Legacy collections

Concurrency, Introduction to threads, Interrupting threads, thread states, thread properties, synchronization, blocking queues, thread safe collections, callable & futures, executors, synchronizers

COURSE OUTCOMES:

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

- Describe the features of Java
- Design classes with object-oriented features
- Describe advanced features of Java like exception handling, multi-threading etc.
- Write programs in JAVA featuring its core capabilities

Text Books:

1. Core Java Volume 1-Fundamentals by Cay S.horstman 10th edition Publisher: Prentice Hall

References:

1. Java the complete reference by Herbert Schield 10th edition Publisher:Tata Mc Graw Hill
2. Java How to program Dietel and Dietel

MSc. IT Syllabus – Department of Computer Science, IUST

List of Practicals:

LAB JOURNAL_ MIT529E	
S. No.	TITLE
1.	Write a program to implement different primitive data types.
2.	Write a program to implement different types of operators in Java.
3.	Write a program to develop a simple cipher in Java using bitwise operators.
4.	Write a program to input two integers from the keyboard and display their sum.
5.	Write a program to implement a class "Test" in Java that defines a static field and a static method to initialize the field. Use another class to create an instance of the class "Test" and access its members using the class name and instance name.
6.	Create a package "MyClasses" and add the class "Test" to the package. Use another class to import the MyClasses package to use the Test class by calling its members.
7.	Write a program to swap two values using object reference. Your program should have a swap function.
8.	Write a class Employee and a class Teacher. Add general properties and methods of employees in the Employee class and add the special properties and methods of teacher in the Teacher class.
9.	Write a Java program that creates a 2D integer array with 5 rows and varying number of columns in each row. Using 'for each' variant of for loop display each element of every row.
10.	Write a Java program that calculates factorial of a number recursively.
11.	WAP that describes a class person. It should have instance variables to record name, age and salary. Create a person object. Set and display its instance variables.
12.	WAP that implements method overloading.
13.	WAP that shows passing object as parameter.
14.	WAP that illustrates method overriding
15.	Write a program to show that the value of non static variable is not visible to all the instances, and therefore cannot be used to count the number of instances.
16.	WAP to illustrate simple inheritance
17.	WAP illustrating a super class variable a referencing as sub class object.
18.	WAP illustrating all uses of super keywords.
19.	Create an abstract class shape. Let rectangle and triangle inherit this shape class. Add necessary functions.
20.	Write a Java program in which a Class overloads a method sum(), which takes 2 parameters. The overloaded methods should perform summation of either integer or floating-point values.
21.	Write a program to print the public fields, constructors, methods their return types and argument of a class using reflection.
22.	Write a java package to show dynamic polymorphism and interfaces.
23.	Write a program to implement a lambda expression with a suitable functional interface that takes two integers and return their sum.
24.	Write at least three interfaces and write at least one default method in any one of the interfaces. Write a class and implement all the interfaces in a class.
25.	Write an application that shows the usage of try, catch, throws and finally.
26.	Write an application that shows how to create a user-defined exception.
27.	Write a class with a method that throws an Arithmetic Exception. Use try catch blocks in the code calling the method to handle the exception.
28.	Write a program to implement a generic class and a generic method that receive different types of data and prints them.
29.	Implement different legacy collections in Java.
30.	Implement stack and queue in Java
31.	Write a program to use Maps collection in Java.

MSc. IT Syllabus – Department of Computer Science, IUST

32.	Write a program to demonstrate the creation and usage of multiple threads in Java.
33.	Write an application that executes two threads. One thread displays —A every 1000 milliseconds and other displays —B every 3000 milliseconds. Create the threads by extending the Thread class.
34.	Write a multithreaded program that demonstrates the problem of race conditions.
35.	Write a program to demonstrate different concurrency control mechanism to avoid race conditions in multithreaded java programs.





Semester - IV

MSc. IT Syllabus – Department of Computer Science, IUST

Semester-IV (24 Credit Semester and 32 hours per week)								
Course Code	Course Name	Paper category	Total Credit	Credit Hours			Hours per week	Course Type
				L	T	P		
18 Core Credit Units (Mandatory)								
MIT651C	Theory of Computation & Formal Languages	Core	4	3	1	0	04	FC
MIT652C	Big Data Analytics	Core	4	3	1	0	04	AEC
MIT653C	Project work	Core	10	1	2	7	17	EOC
06 Discipline Centric Elective Credit Units (One from each pool)								
Pool A: 03 Discipline Centric Elective Credit Units								
MIT661E	Research Methodology	DCE	3	2	1	0	03	FC
MIT662E	Cloud and Grid Computing	DCE	3	2	1	0	03	SEC
MIT663E	Cyber Security	DCE	3	2	1	0	03	SEC
MIT664E	Compiler Design	DCE	3	2	1	0	03	AEC
MIT665E	Bioinformatics	DCE	3	2	1	0	03	AEC
MIT666E	Generative AI	DCE	3	2	1	0	03	AE
Pool B: 03 Discipline Centric Elective Credit Units								
MIT676E	Block Chain Technologies	DCE	3	2	0	2	04	EC
MIT677E	Quantum Computing	DCE	3	2	0	2	04	AEC
MIT678E	Advanced Java	DCE	3	2	0	2	04	EC
MIT679E	Deep Learning	DCE	3	2	0	2	04	SEC
Total			24				32	

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Theory of Computation & Formal Languages	Course Code:	MIT651C
Semester:	4 th	Paper Type:	Core
Credits:	04	Max Marks:	100
Pre Requisite:	MIT615E, MIT503C	Co-Requisite:	MIT664E
Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To be able to construct finite state machines and the equivalent regular expressions.*
- *To understand the class of languages described by finite state machines and regular expressions.*
- *To understand and construct pushdown automata and the equivalent context free grammars.*
- *To understand the concept of construct Linear bounded automata and the equivalent context Sensitive grammars.*
- *To understand the concept of Turing Machine and Universal Turing machine*

COURSE CONTENT:

UNIT I

Introduction to Set Theory, Introduction to Automata, Alphabets, String, Languages, Strings, Kleen Star, Kleen Plus, Introduction to Finite automata, Deterministic finite automata, DFA notations (Transition Table, Transition Table) Languages Accepted by DFA, Designing DFA by pattern recognition, Application of DFA, Non-deterministic Finite Automata, Language accepted by NDFA, Conversion of DFA to NDFA, Overview of e-NDFA. DFA vs NDFA vs e-NDFA.

UNIT II

Finite Automata & Regular Expressions: Definition, Basic Regular Expressions, Obtaining regular expression: Using Basics Expressions & State Elimination Method, Application of regular Expressions, Regular Languages: Definition, Properties, and Pumping Lemma for Regular Languages, Decidable and Closure properties of Regular languages. Limitation of Finite Automata, Equivalence & minimization of DFA.

UNIT III

Introduction to Grammar, Chomsky Hierarchy, Generation of Grammar from Finite Automata & Regular Expressions, Derivation Trees, Left most and Right Most Derivation Trees, Ambiguous Grammar, Context Free Grammar and its applications, properties of context free languages. Introduction to Push down Automata, Transition & Graphical Representation.

UNIT IV

Introduction to Linear bounded Automata & Turing Machine. Turing machine model, transition table Instantaneous definition, Acceptance and Construction of Turing machine. Standard Turing

machine, Transducers, Lexical and Syntax Analyze: Parsing, Top down Parsing (Predictive parsers, LL (K), recursive Descent) and Bottom up Parsing (LR (K)).

COURSE OUTCOMES:

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

- *Describe the mathematical model of machines.*
- *Understand the concept of formal language and corresponding automaton.*
- *Introduces the concept of ambiguity, derivations and parse tree in grammar.*
- *Apply the acquired knowledge of formal languages to the engineering areas like Compiler Design*
- *Apply the acquired knowledge of finite automata theory and to design discrete problems to solve by computers.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. Finite Automata and Formal Languages"- A Simple Approach, A.M Padma Read Pearson Education

References:

1. Hopcroft, J., & Ullman, J., "Introduction to Automata Theory, Languages & Computation", Pearson Education
2. Hopcroft J, R. Motwani, & J. Ullman, "Introduction to Automata Theory, Languages & Computation", Pearson Education
3. P. Linz, "Introduction to Formal Languages & Automata", Jones & Barlett PWS Publishing Company
4. Donald Knuth, "The Art of Computer Programming", Prentice Hall



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Big Data Analytics

Semester: 4th

Credits: 04

Pre Requisite: MIT502C, MIT552C

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

Course Code: MIT652C

Paper Type: Core

Max Marks: 100

Co-Requisite: MIT677E,MIT676E

COURSE OBJECTIVES:

- *To learn big data challenges in different domains including social media*
- *To learn MAP-REDUCE programming model for better scalability and performance*
- *To analyze the capability of No-SQL systems*
- *To apply machine learning algorithms for big data analytics*

COURSE CONTENT:

UNIT I

Introduction to Big Data: Big Data - Definition, overview of Big Data, Big Data Characteristics Volume, Velocity, Variety and other Vs, Issues and challenges of Big Data, Stages of analytical evolution, Big Data Architecture, Space of Big Data: Transactions, Interactions, Observations; Big data Technological approaches and Potential use cases for Big Data. Storage: A General Overview of High-Performance Architecture; HDFS, MapReduce, Map Reduce Programming Model, Overview of HBase, Hive and PIG. Spark Framework – Spark Data frames, Importing and saving data.

UNIT II

Data collection and management: Introduction, Sources of data, Data collection and APIs, data quality and issues with data collection systems with examples from business; Cleaning and treatment of missing data, principles of data visualization, and different methods of presenting data in business analytics. Introduction to Statistics - Measures of central tendencies and distributions, Analysis of variance, Correlation analysis. Distribution properties and arithmetic. Basic analysis techniques - Statistical hypothesis generation and testing.

UNIT III

Machine Learning: Introduction, Supervised Learning, Unsupervised Learning. Classification: Decision Trees - Overview of a Decision Tree, Evaluating a Decision Tree, Confusion matrix. NoSQL Databases Review of traditional Databases, Need for NoSQL Databases, Columnar Databases, Failover and reliability principles, working mechanisms of NoSQL Databases HBase, Cassandra, Couch DB, Mongo DB.

UNIT IV

Data Science - Introduction, Terminology, data science process, data science toolkit. Applications of Data Science, Data Formats – CSV/JSON/AVRO/Parquet. Data Visualization - Technologies for visualization, Tools and libraries for Big Data visualization (e.g., Matplotlib/Seaborn/Tableau/Power-BI). Effective communication of Big Data insights through visualizations.

COURSE OUTCOMES:

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

- *Preparing for data summarization, query, and analysis.*
- *Applying data modelling techniques to large data sets*
- *Creating applications for Big Data analytics*
- *Building a complete business data analytic solution*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. Big Data – A Primer, H. Mohanty, P. Bhuyan, D. Chenthati (Eds.), Springer , Studies in Big Data, vol. 11, 2015.
2. Big Data Analytics with Rand Hadoop, Vignesh Prajapati, PACKT Publishing

References:

1. Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, Cambridge Universities Press, 2012.
2. Big Data at Work: Dispelling the Myths, Uncovering the Opportunities Book by Thomas H. Davenport



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Project Work	Course Code:	MIT653C
Semester:	4 th	Paper Type:	Core
Credits:	10	Max Marks:	250
Pre Requisite:	All courses upto semester 3 rd	Co-Requisite:	MIT-652C, MIT-661E
Marks Distribution: (Mid Term: 100, End Term: 150, Viva: 0, Assignment / Presentations:0)			

COURSE OBJECTIVES:

- *Problem Identification: Identify a relevant problem or challenge in the domain of computer science or information technology.*
- *Research and Analysis: Conduct a thorough literature review and analysis of existing solutions, methodologies, and technologies related to the chosen problem.*
- *Solution Design: Design a novel software solution, algorithm, or framework to address the identified problem or improve existing solutions.*
- *Implementation: Implement the proposed solution using appropriate programming languages, tools, and technologies.*
- *Testing and Validation: Perform rigorous testing and validation to ensure the correctness, reliability, and performance of the developed solution.*
- *Documentation: Prepare comprehensive documentation including project requirements, design specifications, user manuals, and technical reports.*
- *Presentation and Communication: Effectively communicate the project objectives, methodology, findings, and outcomes through presentations, reports, and demonstrations.*
- *Collaboration: Collaborate with peers, faculty mentors, and industry professionals to gather feedback, exchange ideas, and refine the project deliverables.*

COURSE OUTCOMES:

- *Software Prototype: Develop a functional prototype or software application demonstrating the effectiveness and feasibility of the proposed solution.*
- *Research Contribution: Make a valuable contribution to the existing body of knowledge in the field through novel insights, methodologies, or techniques.*
- *Real-world Impact: Address a real-world problem or challenge faced by organizations, industries, or communities, thereby creating tangible benefits and impact.*
- *Technical Skills Enhancement: Enhance technical skills in areas such as software development, algorithm design, data analysis, and project management.*
- *Problem-solving Abilities: Develop critical thinking, problem-solving, and analytical skills by tackling complex problems and finding innovative solutions.*
- *Communication Skills: Improve written and oral communication skills by effectively documenting the project work and presenting findings to diverse audiences.*
- *Professional Development: Gain practical experience, confidence, and readiness for entering the workforce or pursuing further studies in the field of computer science or related areas.*
- *Networking Opportunities: Build professional relationships, network with industry professionals, and explore potential career opportunities through project collaborations and presentations.*



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Research Methodology

Semester: 4th

Credits: 03

Pre Requisite: MIT603C, MIT604C, MIT513E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT661E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT652C, MIT653C

COURSE OBJECTIVES:

- *The objective of this course is to give an introduction to the fundamentals of the wireless communications systems, the wireless network architectures, protocols, and applications.*
- *To provide an overview of Wireless Communication networks area and its applications in communication engineering.*
- *To enable the student to synthesis and analyze wireless and mobile cellular communication systems over a stochastic fading channel*
- *To provide the student with an understanding of advanced multiple access techniques.*
- *To provide the student with an understanding of diversity reception techniques*
- *To give the student an understanding digital cellular systems (cdma2000, W-CDMA, LTE and candidate 5G waveforms).*

COURSE CONTENT:

UNIT I

Wireless Communication Systems: Basic Concepts, Wireless Networks, Wireless vs Wired networks, Classification of Wireless Networks. Wireless Transmission Concepts: Antennas, Types of antennas, Signal propagation, Multipath propagation. Types of Wireless Telephones (Cordless, Fixed Wireless (WLL), Wireless With Limited Mobility (WLL-M), Fully Mobile Wireless Phones). Introduction to various Generations of Mobile Phone Technologies. Wireline/Wireless Portion of Mobile Communication Network.

UNIT II

Electromagnetic spectrum, its use & allocation to well-known bands. Multiple Access Techniques: FDMA, TDMA & CDMA. Basic concepts of Spread Spectrum(SS) technique; Direct Sequence SS versus Frequency Hopping SS. Concept of cells, sectorization, coverage area, frequency reuse, cellular networks, microcells & handoffs. Simplified Implementation of IS-95 CDMA Using Chip Sequences, Concept of (Walsh) Code, Concept of CDMA (PCS and Cellular) Channel, Forward & Reverse CDMA Channel for a Cell/Sector, Comparison of Cellular/PCS CDMA Networks, Frequencies & Cell-Sizes.

UNIT III

Cellular Mobile Systems: Introduction & Overview of GSM/CDMA. Channel: Types, Forward & Reverse CDMA channel for a cell/sector (Pilot, Sync, Paging, Forward Traffic Channels, Access, Reverse TC). Concept of Voice Coding. Components of Mobile Network Architecture: MS, BTS, BSC, MSC, & their basic Functions. Types of Handoffs in GSM. Use of HLR & VLR in Mobile

Networks. Concept of Mobile IP & WAP. Wireless LAN Technology: Overview of Wifi & the IEEE 802.11. Concept of Bluetooth & IEEE 802.15.

COURSE OUTCOMES:

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

- *Demonstrate advanced knowledge of networking and wireless networking and understand various types of wireless networks, standards, operations and use cases.*
- *Demonstrate knowledge of protocols used in wireless networks and learn simulating wireless networks.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. T. Rappaport, “Wireless Communications, Principles and Practice”, Pearson and Dornan, “The Essential Guide to Wireless Communications”, Pearson
2. Jochen Schiller, “Mobile Communications”, Pearson and Andrew Tanenbaum, “Computer Networks”, PHI

References:

3. William Stallings, “Wireless Communications & Networks”, Pearson
4. K.Pahlavan, P.Krishnamurthy, “Principles of Wireless Networks”, Pearson Education.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Cloud & Grid Computing	Course Code:	MIT662E
Semester:	4 th	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT553C, MIT629E, MIT564E	Co-Requisite:	MIT663E, MIT652C
Marks Distribution:	(Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)		

COURSE OBJECTIVE:

- *The objective of studying Cloud and Grid Computing is that it enables new social services by connecting users via social networks that are constructed using multiple cloud service and it will run the software industry upside down. Thus educates potential users about the benefits of cloud computing and the best way to make full use of them.*

COURSE CONTENT:

UNIT I

Cloud Computing: Introduction to cloud & Cluster computing, cloud computing vs cluster computing, Evolution of cloud computing, principles of cloud computing, cloud Computing architecture, Cloud computing applications. Cloud service models (IaaS, PaaS, SaaS), cloud Deployment models (Public, Private, hybrid & community models). Challenge and Security Issues

UNIT II

Grid Computing: Introduction to Grid Computing, Characteristics, grid computing Architecture, grid complexity levels and topologies, grid components and grid layers, applications of Grid Computing. Grid security issues: Authorization and Authentication methods, Grid computing vs Cloud Computing Cluster Computing.

UNIT III

Introduction to OGSA, Services, Schema and architecture, Overview of OGSI, Virtualization: Virtual machines and visualization of clusters and data centres, levels of virtualization, virtualization structures and tools and mechanism, virtualization of CPU, Memory & I/O devices

COURSE OUTCOMES:

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

- *Elaborate the basic concepts of cloud computing and defining the basic terms*
- *Understand various cloud implementations and migration techniques*
- *To define the various industrial applications of cloud virtualization.*
- *learn security challenges and preventive measures in cloud computing*

Text Books:

1. Cloud Computing by pankaj Sharma published by S.K.Kataria and Sons.
2. Cloud Computig Bible:Published by Wiley India Pvt. Ltd.
3. Grid and Cloud Computing and Applications: The 2014 World Comp InternationalConference.Publisher C.S.R.E.A
4. Grid, Cloud and Cluster Computing and Applications by Hamid R.Arabnia, Ashu M.G.Solo

References:

1. Mastering Cloud Computing, Paperback-1 Feb 2013 by Buyya, Vecchiola&Selvi.
2. Cloud Computing: Concepts, Technology & Architecture, 1e Paperback-2014 by Erl.
3. Grid & Cloud Computing, a business perspective on Technology & Applications, Springer by Stanoevska-Slabeva, Katarina, Wozniak, Thomas, Ristol, Santi (Eds.)

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Cyber Security

Semester: 4th

Credits: 03

Pre Requisite: MIT513E, MIT553C, MIT512E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT663E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT662E

COURSE OBJECTIVES:

- *To develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.*
- *To Gain familiarity with prevalent network and distributed system attacks, defenses*
- *To Understand Cybersecurity Fundamentals: Gain a comprehensive understanding of cybersecurity concepts, principles, and best practices.*
- *To Identify Cyber Threats and Vulnerabilities: Learn to recognize and analyze various cyber threats, vulnerabilities, and attack vectors targeting computer systems, networks, and data.*
- *Learn to design, configure, and secure network infrastructure components, including firewalls, intrusion detection/prevention systems (IDS/IPS), and secure communication protocols.*

COURSE CONTENT:

UNIT I

Introduction to Cybersecurity: Overview of cybersecurity concepts and terminology, Cyber threat landscape and evolving cybersecurity trends, Legal and ethical considerations in cybersecurity

Information Security Foundations: Principles of information security: confidentiality, integrity, availability (CIA triad), Risk management: risk assessment, risk analysis, risk mitigation strategies, Security policies, standards, and best practices

UNIT II

Fundamentals of cryptography: encryption, decryption, cryptographic algorithms, Symmetric and asymmetric encryption techniques, Digital signatures, certificates, and public key infrastructure (PKI),

Network Security: Network threats and vulnerabilities, Firewalls, intrusion detection/prevention systems (IDS/IPS), and network segmentation, Secure communication protocols: SSL/TLS, VPNs

Secure coding principles and practices, Common software vulnerabilities: buffer overflow, injection attacks, etc. , Secure software development lifecycle (SDLC)

UNIT III

Web Application Security: Web application threats and vulnerabilities, OWASP Top Ten vulnerabilities, Web application firewalls (WAF), secure coding practices for web development. Identity and Access Management (IAM): Authentication methods: passwords, biometrics, multi-factor authentication (MFA), Authorization and access control models, Identity management systems and single sign-on (SSO) solutions. Emerging Trends in Cybersecurity: Overview of emerging technologies (e.g., IoT, AI/ML) and their security implications, Cybersecurity careers and professional certifications, Future directions and challenges in cybersecurity

COURSE OUTCOMES:

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

- *Demonstrate a comprehensive understanding of cybersecurity concepts, principles, and practices.*
- *Identify and analyze various cyber threats, vulnerabilities, and attack vectors targeting computer systems, networks, and data.*
- *Apply risk management principles and methodologies to assess, analyze, and mitigate cybersecurity risks effectively.*
- *Design, configure, and secure network infrastructure components to safeguard against network-based attacks and unauthorized access.*
- *Assess and mitigate web application vulnerabilities using secure coding practices, web application firewalls (WAF), and other security mechanisms.*

MSc. IT Syllabus – Department of Computer Science, IUST

Text Books:

1. "Cybersecurity Essentials" by Charles J. Brooks and Christopher Grow.

References:

1. "Cryptography and Network Security: Principles and Practices" by William Stallings
2. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto
3. "Security in Computing" by Charles P. Pfleeger, Shari Lawrence Pfleeger, and Jonathan Margulies
4. "The Practice of Network Security Monitoring: Understanding Incident Detection and Response" by Richard Bejtlich



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Compiler Design	Course Code:	MIT664E
Semester:	4 th	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT551C, MIT503C	Co-Requisite:	MIT651C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To understand the various phases in the design of a compiler.*
- *To understand the design of top-down and bottom-up parsers.*
- *To understand syntax directed translation schemes.*
- *To learn to develop algorithms to generate code for a target machine.*

COURSE CONTENT:

UNIT I

Introduction to compilers: Structure of a compiler – Lexical Analysis – Role of Lexical Analyzer – Input Buffering – Specification of Tokens – Recognition of Tokens – Lex – Finite Automata – Regular Expressions to Automata – Minimizing DFA.

UNIT II

Syntax Analysis: Role of Parser – Grammars – Error Handling – Context-free grammars – Writing a grammar – Top Down Parsing – General Strategies Recursive Descent Parser Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR(0)Item Construction of SLR Parsing Table -Introduction to LALR Parser – Error Handling and Recovery in Syntax Analyzer

Intermediate Code Generation: Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking.

UNIT III

Run-Time Environment And Code Generation: Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management— Issues in Code Generation – Design of a simple Code Generator.

Code Optimization: Principal Sources of Optimization – Peep-hole optimization – DAG- Optimization of Basic Blocks Global Data Flow Analysis – Efficient Data Flow Algorithm.

COURSE OUTCOMES:

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

- *Get an understanding of how the different phases of compiler work.*
- *Study of lexical analysis, and various parsing techniques.*
- *Use compiler tools like lex and yacc.*
- *Implement a simple compiler for a hypothetical machine using syntax directed translation*

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson.
2. Compiler Construction-Principles and Practice, Kenneth C Louden, Cengage Learning.

MSc. IT Syllabus – Department of Computer Science, IUST

References:

1. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
2. The Theory and Practice of Compiler writing, J. P. Tremblay and P. G. Sorenson, TMH



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Bioinformatics	Course Code:	MIT665E
Semester:	3 rd	Paper Type:	DCE
Credits:	03	Max Marks:	75
kPre	MIT552C, MIT579E, MIT502C, MIT604C	Co-Requisite:	MIT652C, MIT653C

Requisite:

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To get introduced to the basic concepts of Bioinformatics and its significance in Biological data analysis.*
- *Describe the history, scope and importance of Bioinformatics and role of internet in Bioinformatics.*
- *Explain about the methods to characterize and manage the different types of Biological data.*
- *Classify different types of Biological Databases.*
- *Introduction to the basics of sequence alignment and analysis.*
- *Overview about biological macromolecular structures and structure prediction methods.*

COURSE CONTENT:

UNIT I

Introduction to biology, Living cell: characteristics & functions, basics of molecular biology: DNA, RNA, gene, genetic code, processes of transcription, translation, splicing etc. ,basics of proteins, sequences, structure & functions of proteins, Introduction to bioinformatics: Definitions & concepts, & the role of bioinformatics, Types of biological data.

UNIT II

Introduction to databases, types of Biological Databases – flat file databases, relational databases, object-oriented databases, introduction to biological databases, Sequence databases (EMBL, GenBank, DDBJ, SWISS-PROT, PIR, TrEMBL), search engines: SRS, ENTREZ. BLAST, FASTA.

UNIT III

Basic overview of algorithms: Space & time complexity. Need & significance of sequence alignment. Global & local alignments, techniques of sequence alignments: Doplots, Needleman-wunsch algorithm for global alignment, Smith-waterman algorithm for local alignments, substitution matrices: PAM, Blosum etc. Gene prediction (ab initio & similarity based). Ontologies in bioinformatics: need for ontologies.

COURSE OUTCOMES:

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

- *knowledge and awareness of the basic principles and concepts of biology, computer science and mathematics*
- *Existing software effectively to extract information from large databases and to use this information in computer modeling*
- *Problem-solving skills, including the ability to develop new algorithms and analysis methods*
- *An understanding of the intersection of life and information sciences, the core of shared concepts, language and skills the ability to speak the language of structure-function relationships, information theory, gene expression, and database queries*

Text Books:

1. T. K. Attwood & D J Parry-Smith, "Introduction to bioinformatics", Pearson Education
2. Jean-Michel Claveriw, Cedric Notredame, "Bioinformatics – A beginner's Guide", WILEY DreamTech India Pvt

MSc. IT Syllabus – Department of Computer Science, IUST

References:

1. Harshawardhan P. Bal.” Bioinformatics Principles & Applications” O’Reily
2. T. K. Attwood & D J Parry-Smith, “Introduction to bioinformatics”, Pearson Education
3. Jean-Michel Claveriw, CerdricNotredame, “Bioinformatics – A beginner’s Guide”, WILEY DreamTech India Pvt
4. S.C. Rastogi, N. Mendiratta” Bioinformatics Methods &Applications”TMH
5. Krane ,”Bioinformatic”, Pearson Education
6. B. Bergeron, “Bioinformatics Computing”, Pearson
7. D. E. Krene& M.L. Payma, “Fundamental concepts of Bioinformatics”, Pearson



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Generative AI

Semester: 4th

Credits: 03

Pre Requisite: MIT612E, MIT629E, MIT602C, MIT615E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT666E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT652C

k

COURSE OBJECTIVES:

- *Understand the fundamental principles of Generative AI, including machine learning and deep learning concepts.*
- *Gain practical experience in implementing and training classical and advanced generative models.*
- *Explore the ethical implications and real-world applications of generative AI across various industries.*
- *Develop critical thinking and problem-solving skills through hands-on projects and research discussions.*

COURSE CONTENT:

UNIT I - Introduction to Generative AI

Definition and scope of Generative AI Overview of generative models, Applications & Importance of Generative AI in various domains, Ethical considerations and challenges.

Introduction to language models and their role in AI, Traditional approaches to language modeling Deep learning-based language models and their advantages. Overview of popular LLM architectures: RNNs, LSTMs, and Transformers.

UNIT II - Understanding GPT

Introduction to GPT and its significance. Pre-training and fine-tuning processes in GPT. Architecture and working of GPT models Overview of GPT variants and their use cases. Handling user queries and generating responses. ChatGPT's performance.

UNIT III - Prompt Engineering

Understanding the concept and significance of prompt engineering. Strategies for designing effective prompts Techniques for controlling model behavior and output quality. Best practices for prompt engineering in generative AI.

Introduction to LangChain and its objectives. Overview of the LangChain framework and its components. Streamlining application development using LangChain. Examples of applications built with LangChain.

COURSE OUTCOMES:

- *Demonstrate proficiency in implementing and evaluating generative models for image, text, and sequence generation tasks.*
- *Analyze and critique the strengths and limitations of different generative AI techniques.*
- *Apply ethical frameworks to address biases and fairness issues in generative AI applications.*
- *Communicate effectively about generative AI concepts, methods, and applications through written reports and oral presentations.*

Text Books:

1. Artificial Intelligence & Generative AI for Beginners: The Complete Guide (Generative AI & Chat GPT Mastery Series), WILEY.
2. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, OREILLY.

References:

1. Artificial Intelligence & Generative AI for Beginners: The Complete Guide (Generative AI & Chat GPT Mastery Series), WILEY.
2. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, OREILLY.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title: Block Chain Technologies

Semester: 4th

Credits: 03

Pre Requisite: MIT612E, MIT629E

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

Course Code: MIT676E

Paper Type: DCE

Max Marks: 75

Co-Requisite: MIT652C

COURSE OBJECTIVES:

- To understand the function of Block chain as a method of securing distributed ledgers
- To familiarize the functional/operational aspects of cryptocurrency ecosystem.
- To familiarize about wallets and learn their utilization of wallet during transaction
- To understand that how to write and apply the Smart Contracts
- To understand the concept of Hyperledger

COURSE CONTENT:

UNIT I

Introduction to Cryptography, Introduction to group, ring and field, prime and relative prime numbers, modular arithmetic, Fermat's and Euler's theorem, Euclid's Algorithm, RSA algorithm, Diffie-Hellman key exchange algorithm, ElGamal Encryption, Elliptic curve cryptography, SHA 256, Digital Signature, Zero Knowledge Proof (ZKP)

UNIT II

Introduction from barter system to Cryptocurrency, fundamental of Blockchain, Block structure, Genesis Block, Orphaned Blocks, Stale Block, Uncle Block, Distributed Ledger Technology (DLT), peer-to-peer network, Merkle Tree, Lifecycle of Blockchain, Evolutions of Blockchain, Fork, double spending money, Transactions and UTXO's, Types of Blockchain. Need of Blockchain, Benefits of Blockchain

UNIT III

Cryptocurrencies: BitCoin (BTC), Ethereum (ETH), Ripple (XRP), LiteCoin (LTC), Bitcoin Cash (BCH), Mining pools, Mining, Difficulty Level, Current Target, Nonce, how miners picks transactions, How do mempools work, 51% attack. Consensus Algorithms: Proof of Work (PoW), Asynchronous Byzantine Agreement, Proof of Stake (PoS), Hybrid models (PoW + PoS),

COURSE OUTCOMES:

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

- Implement the blockchain
- Implement the smart contracts on Ethereum platform.
- Implement the use cases on Hyperledger

Text Books

1. Mastering Blockchain, Imran Bashir, Packt Publishing
2. Bitcoin and Cryptocurrency Technologies, Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Princeton University Press. <https://bitcoinbook.cs.princeton.edu/>

References:

1. Grokking Bitcoin, Kalle Rosenbaum, Manning Publications. <http://rosenbaum.se/book/grokking-bitcoin.html>
2. Andrew W. Appel, "Modern compiler implementation in C" Cambridge University Press
3. Blockchain Basics, Daniel Drescher, Apress Publication

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Quantum Computing	Course Code:	MIT677E
Semester:	4 th	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT604C	Co-Requisite:	MIT676E
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *The course will provide an insight of basic of quantum physics from a computer scientist's perspective, and how it describes reality and understand the philosophical implications of quantum computing*

COURSE CONTENT:

UNIT I

Qubit & Quantum States: The Qubit, Vector Spaces. Linear Combination Of Vectors, Uniqueness of a spanning set, basis & dimensions, inner Products, orthonormality, gram-schmidt orthogonalization, bra-ket formalism, the Cauchy-schwarz and triangle Inequalities.

UNIT II

Matrices & Operators: Observables, The Pauli Operators, Outer Products, The Closure Relation, Representation of operators using matrices, outer products & matrix representation, matrix representation of operators in two dimensional spaces, Pauli Matrix, Hermitian unitary and normal operator, Eigen values & Eigen Vectors, Spectral Decomposition, Trace of an operator, important properties of Trace, Expectation Value of Operator, Projection Operator, Positive Operators.

UNIT III

Commutator Algebra, Heisenberg uncertainty principle, polar decomposition & singular values, Postulates of Quantum Mechanics.

COURSE OUTCOMES:

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

- *Achieve knowledge of Vector spaces, Matrices, Quantum state, Density operator and Quantum Measurement theory*

Text Books:

1. Quantum Computing without Magic by Zdzislaw Meglicki
2. Quantum Computing Explained By DAVID Mc MAHON =

References:

1. Quantum Computer Science By Marco Lanzagorta, Jeffrey Uhlmann
2. An Introduction to Quantum Computing Phillip Kaye, Raymond Laflamme, Michele Mosca.

List of Practicals:

LAB JOURNAL_ MIT677E	
S. No.	TITLE
1.	Introduction to Qubits & Vector Spaces Represent qubits using the Dirac notation (bra-ket). Explore the properties of qubits in two-dimensional Hilbert space.
2.	Linear Combination & Spanning Sets Learn about linear combinations of vectors and spanning sets. Determine the uniqueness of a spanning set in a given vector space.

MSc. IT Syllabus – Department of Computer Science, IUST

3.	Basis & Dimensions Identify bases and dimensions of qubit states. Explore the concept of orthonormality and apply the Gram-Schmidt orthogonalization process.
4.	Inner Products & Triangle Inequalities Calculate inner products of qubit states. Apply the Cauchy-Schwarz inequality and triangle inequalities to qubit states.
5.	Observables & Pauli Operators Learn about observables and their representation using Pauli operators. Explore the properties of Pauli operators in quantum mechanics.
6.	Outer Products & Closure Relation Calculate outer products of qubit states. Understand the closure relation in terms of outer products.
7.	Matrix Representation of Operators Learn how to represent operators using matrices. Convert outer products into matrix representations.
8.	Eigenvalues & Eigenvectors Calculate the eigenvalues and eigenvectors of qubit operators. Explore the concept of spectral decomposition in quantum mechanics.
9.	Trace & Expectation Value Calculate the trace of qubit operators and explore its properties. Compute expectation values of qubit operators.
10.	Projection & Positive Operators Understand projection operators and their properties. Explore the concept of positive operators in quantum mechanics.
11.	Commutator Algebra & Heisenberg Uncertainty Principle Learn about commutator algebra and its significance in quantum mechanics. Apply the Heisenberg uncertainty principle to qubit operators.
12.	Polar Decomposition & Singular Values Understand polar decomposition and singular value decomposition in quantum mechanics. Calculate singular values and explore their significance.
13.	Postulates of Quantum Mechanics Recap the postulates of quantum mechanics. Apply the postulates to solve simple quantum mechanics problems.



MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Advanced Java	Course Code:	MIT678E
Semester:	4 th	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT501C, MIT629E	Co-Requisite:	MIT652C, MIT653C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *To learn about streams and how java is using functional programming concepts.*
- *To learn about database connectivity in java.*
- *To learn about socket programming in java.*
- *To learn about Spring Framework in java*

COURSE CONTENT:

UNIT I

Streams: stream operations, stream creation, filter map & flatmap methods, extracting substreams & concatenating streams, other stream transformation, simple reductions, the optional type ,collecting results, collecting into maps, grouping & partitioning, downstream collections, reduction operations, primitive type streams, parallel streams Input & output i /o streams, text input & output, reading & writing binary data, object i/o streams & serialization.

UNIT II

Networking Connecting to a server, implementing servers, Database Programming JDBC design, structured query language, JDBC Configuration, Working with JDBC statements, query execution, scrollable & updatable result sets, row sets, metadata, transactions, advanced sql type, connection management in web & enterprise applications

UNIT III

Core Spring: Introduction to spring, wiring beans, advanced wiring, aspect oriented spring, Spring on web building, spring web application , Spring in the backend using databases with Spring & JDBC, persisting database with object relational mapping, advanced spring MVC, working with spring web flow , securing web applications.

COURSE OUTCOMES:

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

- *Interpret enumerations and collections in advanced Java.*
- *Build programs using collection framework.*
- *Illustrate and develop String Handling methods in JAVA.*
- *Apply Servlets to develop web applications.*
- *Demonstrate database access using JDBC API.*
- *Design reusable software components using JSP.*

Text Books:

1. Core Java Volume 11-Advanced features by Cay S.horstman 10th edition Publisher: Prentice HallSpring in Action, Fourth Edition

References:

1. Pro Spring 5 Authors: Cosmina, I.Harrop, R., Schaefer, C.
2. Spring Boot in Action by Craig Walls

MSc. IT Syllabus – Department of Computer Science, IUST

Course Title:	Deep Learning	Course Code:	MIT679E
Semester:	4 th	Paper Type:	DCE
Credits:	03	Max Marks:	75
Pre Requisite:	MIT612E, MIT602C	Co-Requisite:	MIT652C
Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)			

COURSE OBJECTIVES:

- *The course shall provide a comprehensive understanding of deep learning fundamentals, including neural networks, back propagation, and optimization algorithms.*
- *It will equip students with specialized knowledge in CNNs for image processing, RNNs/LSTMs for sequential data analysis, and practical skills using Tensor Flow and PyTorch.*

COURSE CONTENT:

UNIT I

Introduction to Neural Networks: Understanding Perceptrons, Activation Functions - Sigmoid, ReLU, Tanh, Feedforward Neural Networks, Backpropagation, Algorithm, Loss Functions (e.g., Mean Squared Error, Cross-Entropy), Gradient Descent and Stochastic Gradient Descent, Regularization Techniques (e.g., Dropout, L2 Regularization), Optimization Algorithms (e.g., Adam, RMSprop), Model Evaluation and Metrics, Practical Implementation (e.g., using TensorFlow, PyTorch).

UNIT II

Convolutional Neural Networks (CNNs): Introduction to Image Processing with Neural Networks, Convolutional Layers, Pooling Layers (e.g., Max Pooling, Average Pooling), Padding and Stride, CNN Architectures (e.g., LeNet, AlexNet, VGG, ResNet), Transfer Learning and Fine-Tuning, Object Detection and Localization (e.g., YOLO, SSD), Image Segmentation (e.g., U-Net, FCN), Interpretability and Visualization Techniques, Advanced Topics (e.g., Attention Mechanisms in CNNs)

UNIT III

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM): Introduction to Sequential Data Processing, Basics of Recurrent Neural Networks (RNNs), Vanishing Gradient Problem in RNNs, Long Short-Term Memory (LSTM) Networks, Gated Recurrent Units (GRUs), Bidirectional RNNs Sequence-to-Sequence Models, Attention Mechanisms in RNNs, Applications in Natural Language Processing (NLP), Time Series Analysis and Forecasting with RNNs/LSTMs.

COURSE OUTCOMES:

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

- *Acquire a comprehensive understanding of deep learning concepts, including neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM).*
- *Gain proficiency in implementing deep learning models for image processing, natural language processing (NLP), and time series analysis using popular frameworks such as Tensor Flow and PyTorch.*
- *Develop the skills necessary to apply advanced deep learning techniques to real-world problems, including object detection, image segmentation, and sequence prediction, while also understanding interpretability and visualization methods.*

Text Books:

1. Deep Learning/by Bengio Yoshua .,Courville Aaron.,Goodfellow Ian .Publisher Information : The MIT Press, London: 2016

NEP 2020 Based Focus Areas



Course Outline MSc.IT Semester I

Course Code	Course Name	Course Type
MIT501C	Programming Concepts in C/C++	SEC
MIT502C	Database Management System	EC
MIT503C	Discrete Mathematics	FC
MIT504C	Computer Organization and Architecture	FC
MIT511E	Digital Electronics	SEC
MIT512E	Green Technologies	VAC
MIT513E	Management Information System	EC
MIT526E	Web Technologies	EOC
MIT527E	E-Commerce	EC
MIT528E	Multimedia Technologies	EC
MIT541A	Information and Communications	AEC

Course Outline MSc.IT Semester II

Course Code	Course Name	Course
MIT551C	Data Structures	FC
MIT552C	Introduction to Python	SEC
MIT553C	Data Communication & Computer Networks	FC
MIT554C	Operating System	FC
MIT561E	Web Mining	SEC
MIT562E	Quantitative Techniques	AEC
MIT563E	Microprocessor & Assembly Language Programming	AEC
MIT564E	Internet of Things	SEC
MIT576E	Open Source Technologies	EOC
MIT577E	Computer Graphics	SEC
MIT578E	Numerical and Statistical Computing	VAC
MIT579E	Pervasive Computing	AEC
MIT591A	Technical Communication	VAC

Course Outline MSc.IT Semester III

Course Code	Course Name	Course Type
MIT601C	Full-Stack Development	EC
MIT602C	Artificial Intelligence & Machine Learning	AEC
MIT603C	Software Engineering	FC
MIT604C	Design and Analysis of Algorithms	FC
MIT611E	Mobile Communication	AEC
MIT612E	Distributed Databases	EC
MIT613E	Cryptography and Network Security	SEC
MIT614E	System Software Design	AEC
MIT615E	Natural Language Processing	AEC
MIT626E	Digital Image Processing	AEC
MIT627E	Object Oriented Analysis & Design	EC
MIT628KE	Data Warehouse and Mining	EC
MIT629E	Java Programming	EC

Course Outline MSc.IT Semester IV

Course Code	Course Name	Course Type
MIT651C	Theory of Computation & Formal Languages	FC
MIT652C	Big Data Analytics	AEC
MIT653C	Project work	EOC
MIT661E	Research Methodology	FC
MIT662E	Cloud and Grid Computing	SEC
MIT663E	Cyber Security	SEC
MIT664E	Compiler Design	AEC
MIT665E	Bioinformatics	AEC
MIT666	Generative AI	AE
MIT676E	Block Chain Technologies	EC
MIT677E	Quantum Computing	AEC
MIT678E	Advanced Java	EC
MIT679E	Deep Learning	SEC

Course Summary

Foundation Courses	9
Skill Enhancement Courses	12
Value Added Courses	3
Ability Enhancement Courses	13
Employability Courses	11
Entrepreneurship Oriented Courses	3

