Shri Vile Parle Kelavani Mandal's

Narsee Monjee College of Commerce and Economics

(Autonomous)

	S.Y BSc IT (To be implemented for the academic year 2021-2022)			
Semester III (Total Credits 24)				
Sr. No.	**Module Code**	**Module Name**		**Credits**
1	**NMUBSCIT301**	Python Programming	Skill Enhancement Courses (SEC)	4
2	**NMUBSCITP301**	Python Programming Practical		1
3	**NMUBSCIT306**	Data Structures and Algorithms	Core Course (CC)	4
4	**NMUBSCITP306**	Data Structures and Algorithms Practical		1
5	**NMUBSCIT303**	Computer Networks		3
6	**NMUBSCITP303**	Computer Networks Practical		1
7	**NMUBSCIT304**	Data Base Management System		4
8	**NMUBSCITP304**	Database Management Systems Practical		1

9	**NMUBSCIT307**	Mathematics for Information Technology I		4
10	**NMUBSCITP307**	Mathematics for Information Technology I Practical		1
	Total credits			**24**

Programme : B. Sc. IT (Information Technology)					**Semester : III**		
Course : Python Programming					**Code : NMUBSCIT301**		
Suggested Lectures per week				**04**			
Practical Session per week (per Batch)				**01**			
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Component 1**	**Component 2**
60	**15 X 2 = 30**	**Nil**	**04+01 = 05**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**
Internal Component (Theory Break up)							
Class Test Duration 20 Mins				**Assignments**			
10 Marks				**15 Marks**			
Internal Component (Practical Break up)							

Examination (Duration 1 ½ Hrs)	**Mini Project/Case study/** **Field Visit (report to be submitted and certified prior to practical examination)**
30 Marks	20 Marks

Learning Objectives :

1. Writing programs in Python
2. Using various data structures of python
3. Understanding database connectivity through Python

Learning Outcomes :

Programmers level proficiency in python

1. Developing small project / Analysing existing projectsin python.

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	**Introduction:** The Python Programming Language, History, features, Installing Python, Running Python program, Debugging: Syntax Errors, Runtime Errors, Semantic Errors, Experimental Debugging, Formal and Natural Languages, The Difference Between Brackets, Braces, and Parentheses **Variables and Expressions:** Values and Types, Variables, Variable Names and Keywords, Type conversion, Operators and Operands, Expressions, Interactive Mode and Script Mode, Order of Operations. **Conditional Statements:** if, if-else, nested if –else **Looping**: for, while, nested loops **Control statements:** Terminating loops, skipping specific conditions	Classroom learning Presentation	12

II	**Functions:** Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters Are Local, Stack Diagrams, Fruitful Functions and Void Functions, Why Functions? Importing with from, Return Values, Incremental Development, Composition, Boolean Functions, More Recursion, Leap of Faith, Checking Types **Strings:** A String Is a Sequence, Traversal with a for Loop, String Slices, Strings Are Immutable, Searching, Looping and Counting, String Methods, The in Operator, String Comparison, String Operations.	Classroom learning Presentation	12

III	**Lists:** Values and Accessing Elements, Lists are mutable, traversing a List, Deleting elements from List, Built-in List Operators, Concatenation, Repetition, In Operator, Built- in List functions and methods **Tuples and Dictionaries:** Tuples, accessing values in Tuples, Tuple Assignment, Tuples as return values, Variable-length argument tuples, Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built- in Tuple Functions Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary	Classroom learning Presentation	12
	Functions, Built-in Dictionary Methods Files: Text Files, The File Object Attributes, Directories Exceptions: Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions		
IV	Regular Expressions – Concept of regular expression, various types of regular expressions, using match function. Classes and Objects: Overview of OOP (Object Oriented Programming), Class Definition, Creating Objects, Instances as Arguments, Instances as return values, Built- in Class Attributes, Inheritance, Method Overriding, Data Encapsulation, Data Hiding Multithreaded Programming: Thread Module, creating a thread, synchronizing threads, multithreaded priority queue Modules: Importing module, Creating and exploring modules, Math module, Random module, Time module	Classroom learning Presentation	12

V	Creating the GUI Form and Adding Widgets: Widgets: Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menubutton, Menu, Message, Radiobutton, Scale, Scrollbar, text, Toplevel, Spinbox, PanedWindow, LabelFrame, tkMessagebox. Handling Standard attributes and Properties of Widgets. Layout Management: Designing GUI applications with proper Layout Management features. Look and Feel Customization: Enhancing Look and Feel of GUI using different appearances of widgets. Storing Data in Our MySQL Database via Our GUI: Connecting to a MySQL database from Python, Configuring the MySQL connection, Designing the Python GUI database, Using the INSERT command, Using the UPDATE command, Using the DELETE command, Storing and retrieving data from MySQL database**.**	Classroom learning Presentation	12

<u>List of Practical Questions:</u>

1. Write the program for the following:
   a. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.
   b. Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.
   c. Write a program to generate the Fibonacci series.
   d. Write a function that reverses the user defined value.
   e. Write a function to check the input value is Armstrong and also write the function for Palindrome.
   f. Write a recursive function to print the factorial for a given number.

2. Write the program for the following:
   a. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.
   b. Define a function that computes the length of a given list or string.
   c. Define a procedure histogram () that takes a list of integers and prints a histogram to the screen. For example, histogram ([4, 9, 7]) should print the following: ****

```
*********
*******
```

3. Write the program for the following:

a. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.

b. Take a list, say for example this one:

a= [1,1,2,3,5,8,13,21,34,55,89] and write a program that prints out all the elements of the list that are less than 5.

4. Write the program for the following:

a. Write a program that takes two lists and returns True if they have at least one common member.

b. Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.

c. Write a Python program to clone or copy a list

5. Write the program for the following:

a. Write a Python script to sort (ascending and descending) a dictionary by value.

b. Write a Python script to concatenate following dictionaries to create a new one.

Sample Dictionary: dic1= {1:10, 2:20} dic2= {3:30, 4:40} dic3= {5:50,6:60}

Expected Result: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

c. Write a Python program to sum all the items in a dictionary.

6. Write the program for the following:

a. Write a Python program to read an entire text file.

b. Write a Python program to append text to a file and display the text.

c. Write a Python program to read last n lines of a file.

7. Write the program for the following:

a. Design a class that store the information of student and display the same

b. Implement the concept of inheritance using python

c. Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).

i. Write a method called add which returns the sum of the attributes x and y.

ii. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.

iii. Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.

iv. Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y.

8. Write the program for the following:

a. Open a new file in IDLE ("New Window" in the "File" menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the "Control Flow and Functions" exercise into this file and save it. Now open a new file and save it in the same directory. You should now be able to import your own module like this: import geometry Try and add print dir(geometry) to the file and run it.

Now write a function pointyShapeVolume (x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is False. x is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is False. y is the height of the object. First use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas.

b. Write a program to implement exception handling.

9. Write the program for the following:

a. Try to configure the widget with various options like: bg=" red", family=" times", size=18

b. Try to change the widget type and configuration options to experiment with other widget types like Message, Button, Entry, Checkbutton, Radiobutton, Scale etc.

10. Design the database applications for the following:

a. Design a simple database application that stores the records and retrieve the same.

b. Design a database application to search the specified record from the database.

c. Design a database application to that allows the user to add, delete and modify the records.

Books and References:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**

1.	Think Python	Allen Downey	O’Reilly	1st	2012
2.	An Introduction to Computer Science using Python 3	JasonMontojo, Jennifer Campbell, Paul Gries	SPD	1st	2014
3.	Python GUI Programming Cookbook	Burkhard A. Meier	Packt		2015
4.	Introduction to Problem Solving with Python	E. Balagurusamy	TMH	1st	2016
5.	Murach’s Python programming	Joel Murach, Michael Urban	SPD	1st	2017
6.	Object-oriented Programming in Python	Michael H. Goldwasser, David Letscher	Pearson Prentice Hall	1st	2008
7.	Exploring Python	Budd	TMH	1st	2016

Programme : B. Sc. IT (Information Technology)					**Semester : III**		
Course : Data Structures and Algorithms					**Code : NMUBSCIT306**		
Suggested Lectures per week				**04**			
Practical Session per week (per Batch)				**01**			
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Component 1**	**Component 2**
60	**15 X 2 = 30**	**Nil**	**04+01=5**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**
Internal Component (Theory Break up)							
Class Test Duration 20 Mins				**Assignments**			
10 Marks				**15 Marks**			
Internal Component (Practical Break up)							
Examination (Duration 1 ½ Hrs)				**Mini Project/Case study/ Field Visit (report to be submitted and certified prior to practical examination)**			
30 Marks				20 Marks			
Learning Objectives :							

1. Learning Linear static Data structures and algorithmic concepts
2. Understanding dynamic data structures and their applications
3. Learning Non Linear Data structures
4. Learning hashing techniques

Learning Outcomes :

1. Ability to compare and analyze algorithms in the basis of time and space complexity
2. Ability to implement expression solver, convertor
3. Ability to implement different sorting and searching logics
4. Ability to implement shortest path and similar applications of graph

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	**Introduction:** Data and Information, Data Structure, Classification of Data Structures, Primitive DataTypes, Abstract Data Types, Data structure vs. File Organization,Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm, Asymptotic Analysis and Notations, Big O Notation, Big Omega Notation, Big Theta Notation, Rate of Growth and Big O Notation. **Array**: Introduction, One Dimensional Array, Memory Representation of One Dimensional Array, Traversing, Insertion, Deletion, Searching, Sorting, Merging of Arrays, Multidimensional Arrays, Memory Representation of Two Dimensional Arrays, General Multidimensional Arrays, Sparse Arrays, SparseMatrix, Memory Representation of Special kind of Matrices, Advantages and Limitations of Arrays.	Classroom learning	12

II	**Linked List:** Linked List, One-way Linked List, Traversal of Linked List, Searching, Memory Allocation and Deallocation, Insertion in Linked List, Deletion from Linked List, Copying a List into Other List, Merging Two Linked Lists, Splitting a List into Two Lists, Reversing One way linked List, Circular Linked List, Applications of Circular Linked List, Two way Linked List, Traversing a Two way Linked List, Searching in a Two way linked List, Insertion of an element in Two way Linked List, Deleting a node from Two way Linked List, Header Linked List, Applications of the Linked list, Representation of Polynomials, Storage of Sparse Arrays, Implementing other Data Structures. Complexity analysis of Link List	Classroom learning.	12
III	**Stack**: Introduction, Operations on the Stack Memory Representation of Stack, Array Representation ofStack, Applications of Stack, Evaluation of Arithmetic Expression, Matching Parenthesis, infix and postfix operations, Recursion**.** **Queue:** Introduction, Queue, Operations on the Queue, Memory Representation of Queue, Array representation of queue, Linked List Representation of Queue, Circular Queue, Some special kinds of queues, Deque, Priority Queue, Application of Priority Queue, Applications of Queues. Complexity analysis of stack and queue	Classroom learning.	12
IV	**Tree**: Tree, Binary Tree, Properties of Binary Tree, Memory Representation of Binary Tree, Operations Performed on Binary Tree, Reconstruction of Binary Tree from its Traversals, Binary Search Tree, Operations on Binary Search Tree, Heap, Memory Representation of Heap, Operation on Heap, Heap Sort, B-Tree **Graph**: Introduction, Graph, Graph Terminology, Memory Representation of Graph, Adjacency Matrix Representation of Graph, Adjacency List or Linked Representation of Graph, Operations Performed on Graph, Graph Traversal, Applications of the Graph, Reachability, Shortest Path Problems, Spanning Trees.	Classroom learning.	12

V	Sorting and Searching Techniques Bubble, Selection, Insertion, Merge Sort. Searching: Sequential, Binary, Indexed Sequential Searches, Binary Search. Hashing Techniques Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Buckethashing, Deletion and rehashing. Complexity comparison of different sorting algorithms	Classroom learning	12

List of Practical Questions:

1. Implement the following:
   a. Write a program to store the elements in 1-D array and perform theoperations like searching, sorting and reversing the elements. [Menu Driven]
   b. Read the two arrays from the user and merge them and display the elements in sorted order. [Menu Driven]
   c. Write a program to perform the Matrix addition, Multiplication and Transpose Operation. [Menu Driven]

2. Implement the following for Linked List:
   a. Write a program to create a single linked list and display the nodeelements in reverse order.
   b. Write a program to search the elements in the linked list and display thesame
   c. Write a program to create double linked list and sort the elements in the linked list.

3. Implement the following for Stack:
   .
   a. Write a program to convert an infix expression to postfix and prefix conversion.

4. Implement the following for Queue:
   a. Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations.
   b. Write a program to implement the concept of Circular Queue
   .

5. Implement the sorting techniques:
   a. Write a program to implement selection sort.
   b. Write a program to implement insertion sort.

6. Implement Binary Search

7. Implement the following data structure techniques:
    a. Write a program to create the tree and display the elements.
    b. Write a program to construct the binary tree.
    c. Write a program for inorder, postorder and preorder traversal of tree

8. Implement the following data structure techniques:
    a. Write a program to insert the element into maximum heap.
    b. Write a program to insert the element into minimum heap.

9. Implement the following data structure techniques:
    a. Write a program to implement the collision technique.
    b. Write a program to implement the concept of linear probing.

10. Implement the following data structure techniques:
    a. Write a program to generate the adjacency matrix.
    b. Write a program for shortest path algorithm.

Books and References:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Data Structures and Algorithms Using Python	RanceNecaise	Wiley	First	2016
2.	Data Structures Using C and C++	Langsam,Augenstein, Tanenbaum	Pearson	First	2015

Programme : B. Sc. IT (Information Technology)					**Semester : III**		
Course : Computer Networks					**Code : NMUBSCIT303**		
Suggested Lectures per week				**03**			
Practical Session per week (per Batch)				**01**			
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Component 1**	**Component 2**
45	**15 X 2 = 30**	**Nil**	**03+01 = 04**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**
Internal Component (Theory Break up)							
Class Test Duration 20 Mins				**Assignments**			
10 Marks				**15 Marks**			
Internal Component (Practical Break up)							
Examination (Duration 1 ½ Hrs)				**Mini Project/Case study/ Field Visit (report to be submitted and certified prior to practical examination)**			
30 Marks				20 Marks			

Learning Objectives :

1. To understand networking concepts
2. To understand transmission media and media access protocols
3. To understand Internet Protocol

Learning Outcomes :

1. Ability to understand role of various network levels
2. Ability to operate and work with transmission media
3. Ability to work on Internet Protocol

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	**Introduction:** Data communications, networks, network types, standards and administration. **Network Models:** Protocol layering, TCP/IP protocol suite, The OSI model. **Introduction to Physical layer:** Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance.	Classroom learning /Presentation	09
II	**Bandwidth Utilization: Multiplexing and Spectrum Spreading:** Multiplexing, Spread Spectrum **Transmission media:** Guided Media**,** Unguided Media **Switching:** Introduction, circuit switched networks, packet switching, structure of a switch. **Introduction to the Data Link Layer:** Link layer addressing, Data Link Layer Design Issues, Error detection and correction, block coding, cyclic codes, checksum, forward error correction, error correcting codes, error detecting codes.	Classroom learning /Presentation	09

III	**Data Link Control:** DLC services, data link layer protocols, Point-to-point protocol. **Media Access Control**: Random access, controlled access, channelization, Wired LANs – Ethernet Protocol, standard Ethernet, fast Ethernet, gigabit Ethernet, 10 gigabit Ethernet, **Wireless LANs:** Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks. **Connecting devices and Virtual LANs.**	Classroom learning /Presentation.	09
IV	**Introduction to the Network Layer**: Network layer services, packet switching, network layer performance, IPv4 addressing, forwarding of IP packets, Internet Protocol, ICMPv4, Mobile IP **Unicast Routing:** Introduction, routing algorithms, unicast routing protocols. **Next generation IP:** IPv6 addressing, IPv6 protocol, ICMPv6 protocol, transition from IPv4 to IPv6.	Classroom learning /Presentation	09
V	**Introduction to the Transport Layer**: Introduction, Transport layer protocols (Simple protocol, Stop-and- wait protocol, Go-Back-n protocol, Selective repeat protocol, Bidirectional protocols), Transport layer services, User datagram protocol, Transmission control protocol, **Standard Client Server Protocols:** World wide-web and HTTP, FTP, Electronic mail, Telnet, Secured Shell, Domain name system.	Classroom learning /Presentation	09

List of Practical Questions:

1. IPv4 Addressing and Subnetting
   a) Given an IP address and network mask, determine other information about the IP address such as:
   - Network address
   - Network broadcast address
   - Total number of host bits
   - Number of hosts

   b) Given an IP address and network mask, determine other information about the IP address such as:
   - The subnet address of this subnet
   - The broadcast address of this subnet
   - The range of host addresses for this subnet
   - The maximum number of subnets for this subnet mask
   - The number of hosts for each subnet
   - The number of subnet bits
   - The number of this subnet
2. Use of ping and tracert / traceroute, ipconfig / ifconfig, route and arp utilities.

3. Configure IP static routing.
4. Configure IP routing using RIP.
5. Configuring Simple OSPF.
6. Configuring DHCP server and client.
7. Create virtual PC based network using virtualization software and virtual NIC.
8. Configuring DNS Server and client.
9. Configuring OSPF with multiple areas.
10. Use of Wireshark to scan and check the packet information of following protocols
    - HTTP
    - ICMP
    - TCP
    - SMTP
    - POP3

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Data Communication and Networking	Behrouz A. Forouzan	Tata McGraw Hill	Fifth Edition	2013
2.	TCP/IP Protocol Suite	Behrouz A. Forouzan	Tata McGraw Hill	Fourth Edition	2010
3.	Computer Networks	Andrew Tanenbaum	Pearson	Fifth	2013

Programme : B. Sc. IT (Information Technology)					**Semester : III**		
Course : Database Management System					**Code : NMUBSCIT304**		
Suggested Lectures per week				**04**			
Practical Session per week (per Batch)				**01**			
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Component 1**	**Component 2**
60	**15 X 2 = 30**	**Nil**	**04+01 = 05**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**
Internal Component (Theory Break up)							
Class Test Duration 20 Mins				**Assignments**			
10 Marks				**15 Marks**			
Internal Component (Practical Break up)							
Examination (Duration 1 ½ Hrs)				**Mini Project/Case study/ Field Visit (report to be submitted and certified prior to practical examination)**			
30 Marks				20 Marks			

Learning Objectives :

1. To learn the concept of database and ERD
2. To learn about Relational Model and Normalisation
3. To learn basic queries in SQL
4. To learn advanced queries in SQL and basic queries in PL/SQL **5.** To learn advanced queries inPl/SQL

Learning Outcomes :

1. Able to understand Database Models and Database Design
2. Understand the concept of Normalisation
3. Ability to solve queries in SQL and PL/SQL

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	Introduction to Databases What is database system, purpose of database system, view of data, Database Languages, Database Engine, database architecture, Database users and administrators, The evolution of data models Introduction to the Relational Model Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational algebra	Classroom learning presentation	12

II	Database Design Overview of the Design Process, The Entity-Relationship Model, Complex Attributes, Mapping, Primary Key, Removing Redundant Attributes in Entity Sets, Reducing E-R Diagrams to Relational Schemas, Extended E-R Features, EntityRelationship Design Issues Codd's Rules Normalisation Atomic domain and Normalization (1NF, 2NF, 3NF, BCNF). Denormalization	Classroom learning presentation	12
III	Introduction to SQL SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database Intermediate SQL Join Expressions, Views, comparison between tables and views	Classroom learning presentation	12
IV	SQL Transactions, Integrity Constraints, SQL Data Types and Schemas, Index Definition in SQL, Authorization, Triggers PL/SQL Introduction to PL/SQL, Language Fundamentals, Conditional and Sequential Control, Iterative Processing with Loops	Classroom learning presentation	12
V	PL/SQL Exception Handlers., Records, Collections, Cursors, Procedures, Functions, Packages, Triggers, Transaction Management, Sequences, With Clause and Hierarchical Retrieval	Classroom learning presentation	12

List of Practical

1. For given scenario

   . Draw E-R diagram and convert entities and relationships to table.

2. Perform the following:

   · Viewing all databases
   · Creating a Database
   · Viewing all Tables in a Database
   · Creating Tables (With and Without Constraints)
   · Inserting/Updating/Deleting Records in a Table
   · Saving (Commit) and Undoing (rollback)

3. Perform the following:

   · Altering a Table

· Dropping/Truncating/Renaming Tables
· Backing up / Restoring a Database

4. Perform the following:
   · Simple Queries
   · Simple Queries with Aggregate functions
   · Queries with Aggregate functions (group by and having clause)
5. Queries involving
   · Date Functions
   · String Functions
   · Math Functions
6. Join Queries
   · Inner Join
   · Outer Join
7. Views
   · Creating Views (with and without check option)
   · Dropping views
   · Selecting from a view

8. PL/SQL Practical
   Declaring Variables
   Writing Executable Statements
   Interacting with the Oracle Server
9. Writing Control Structures

   Composite data types, cursors and exceptions.
   Working with Composite Data Types
   Writing Explicit Cursors
   Handling Exceptions

10. Procedures and Functions
    Creating Procedures
    Creating Functions
    Managing Subprograms
    Creating Packages

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Database System and Concepts	A Silberschatz, H Korth, S Sudarshan	McGraw-Hill	Seventh Edition	

2.	Database Systems	RobCoronel	Cengage Learning	Twelfth Edition	
3.	Programming with PL/SQL forBeginners	H.Dand, R.Patil and T. Sambare	X –Team	First	2011
4.	Introduction to Database System	C.J.Date	Pearson	First	2003
5	PL/SQL Programming	Ivan Bayross	BPB	First	2010

Programme : B. Sc. IT (Information Technology)					**Semester : III**		
Course : Mathematics for Information Technology I					**Code : NMUBSCIT307**		
Suggested Lectures per week				**04**			
Practical Session per week (per Batch)				**01**			
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Component 1**	**Component 2**
60	**15 X 2 = 30**	**Nil**	**04+01 = 5**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**
Internal Component (Theory Break up)							
Class Test Duration 20 Mins				**Assignments**			
10 Marks				**15 Marks**			
Internal Component (Practical Break up)							
Examination (Duration 1 ½ Hrs)				**Mini Project/Case study/ Field Visit (report to be submitted and certified prior to practical examination)**			
30 Marks				20 Marks			

Learning Objectives :

1. To understand complex number system, vectors and matrices
2. To understand dimensions and ranks
3. To understand basics of statistics “ Mean mode and central tendency
4. To learn concept of standard deviation
5. To learn R programming language and functions there in.

Learning Outcomes :

1. Ability to solve complex number systems and vector / matrix system
2. Ability to implement concepts of linear algebra
3. Understanding of various statistical concepts and their implementation using R – Programming Language

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	Field: Introduction to complex numbers, Abstracting over fields, Playing with GF(2), Vector Space: Vectors are functions, Vector addition, Scalar-vector multiplication, Combining vector addition and scalar multiplication, Dictionarybased representations of vectors, Dot-product, Solving a triangular system of linear equations. Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise. Matrix: Matrices as vectors, Transpose, Matrixvector and vector-matrix multiplication in terms of linear combinations, Matrix-vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrixmatrix multiplication, Innerproduct and outer product, From function inverse to matrix inverse	Classroom learning	12

II	Basis: Coordinate systems, Two greedy algorithms for finding a set of generators, Minimum Spanning Forest and GF(2), Linear dependence, Basis , Unique representation, Change of basis, first look, Computational problems involving finding a basis Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator. Gaussian elimination: Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination, Finding a basis for the null space, Factoring integers, Inner Product: The inner product for vectors over thereals, Orthogonality, Orthogonalization: Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, Orthogonal complement, Eigenvector: Modeling discrete dynamic processes, Diagonalization of the Fibonacci matrix, Eigenvalues and eigenvectors, Coordinate representation in terms of eigenvectors, The Internet worm, Existence of eigenvalues, Markov chains, Modeling a web surfer: PageRank.	Classroom learning.	12

III	The Mean, Median, Mode, and Other Measures of Central Tendency: Index, or Subscript, Notation, Summation Notation, Averages, or Measures of Central Tendency ,The Arithmetic Mean , The Weighted Arithmetic Mean ,Properties of the Arithmetic Mean ,The Arithmetic Mean Computed from Grouped Data ,The Median ,The Mode, The Empirical Relation Between the Mean, Median, and Mode, The Geometric Mean G, The Harmonic Mean H ,The Relation Between the Arithmetic, Geometric, and Harmonic Means, The Root Mean Square, Quartiles, Deciles, and Percentiles, Software and Measures of Central Tendency	Classroom learning.	12
IV	The Standard Deviation and Other Measures of Dispersion: Dispersion, or Variation, The Range, The Mean Deviation, The Semi-Interquartile Range, The 10–90 Percentile Range, The Standard Deviation, The Variance, Short Methods for Computing the Standard Deviation, Properties ofthe Standard Deviation, Charlie's Check, Sheppard's Correction for Variance, Empirical Relations Between Measures of Dispersion, Absolute and Relative Dispersion; Coefficient ofVariation, Standardized Variable; Standard Scores,Software and Measures of Dispersion.	Classroom learning.	12
V	Introduction to R: Basic syntax, data types, variables, operators, control statements, Rfunctions, R –Vectors, R – lists, R Arrays. Basic R Programming – Fetching external data intoR. Graphics in R.	Classroom learning	12

List of Practical (Use R/ Scilab/ Python)

1. Write a program which demonstrates the following:
   - Addition of two complex numbers
   - Displaying the conjugate of a complex number
   - Plotting a set of complex numbers

- Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number a=1/2, a=1/3, a=2 etc.

2. Write a program to do the following:
   - Enter a vector u as a n-list
   - Enter another vector v as a n-list
   - Find the vector au+bv for different values of a and b Find□ the dot product of u and v 3.
3. Write a program to do the following:
   - Enter two distinct faces as vectors u and v.
   - Find a new face as a linear combination of u and v i.e. au+bv for a and bin R.
   - Find the average face of the original faces.
4. Write a program to do the following:
   - Enter an r by c matrix M (r and c being positive integers)
   - Display M in matrix format
   - Display the rows and columns of the matrix M □ Find the scalar multiplication of M for a given scalar.
   - Find the transpose of the matrix M.
5. Write a program to do the following:
   - Find the vector –matrix multiplication of a r by c matrix M with an cvectoru.
   - Find the matrix-matrix product of M with a c by p matrix N.
6. Write a program to enter a matrix and check if it is invertible. If the inverse exists,find the inverse. 7. Write a program to convert a matrix into its row echelon form
7. Write a program to do the following:

- Enter a positive number N and find numbers a and b such that a2 – b 2 = N □ Find the gcd of two numbers using Euclid's algorithm.

8. Write a program to do the following:

- Enter a vector b and find the projection of b orthogonal to a given vector u.
- Find the projection of b orthogonal to a set of given vectors 10.

9. Write a program to enter a given matrix and an eigen value of the same. Find its eigen vector

10. Using R execute the basic commands, array, list and frames.
11. Create a Matrix using R and Perform the operations addition, inverse, transposeand multiplication operations.
12. Using R Execute the statistical functions: mean, median, mode, quartiles, range, inter quartile range histogram
13. Using R import the data from Excel / .CSV file and Perform the above functions.
14. Using R import the data from Excel / .CSV file and Calculate the standarddeviation, variance, co-variance.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Coding the Matrix Linear Algebra through Applications to Computer Science (Unit I & II)	, PHILIP N. KLEIN,	Newtonian Press	Edition 1	2013
2.	STATISTICS (Unit III, IV)	Murray R . Spiegel, Larry J.Stephens.	McGRAW – HILL ITERNATIONAL	FOURTH	
3.	A Practical Approach using R (Unit V)	R.B. Patil, H.J. Dand and R. Bhavsar	SPD	1st	2017

S.Y BSc IT (To be implemented for the academic year2021-2022)

Semester IV (Total Credits 24)

Sr. No.	Module Code	Module Name		Credits
1	**NMUBSCIT401**	Core Java	***Core Courses(CC)***	4
2	**NMUBSCITP401**	Core Java Practical		1
3	**NMUBSCIT406**	Embedded System	***Skill Enhancement Courses (SEC)***	4
4	**NMUBSCITP406**	Embedded System Practical		1
5	**NMUBSCIT407**	Mobile Programming		3
6	**NMUBSCITP407**	Mobile Programming Practical		1
7	**NMUBSCIT404**	Software Engineering		4
8	**NMUBSCITP404**	Software Engineering Practical		1
9	**NMUBSCIT408**	Mathematics for Information Technology II		4
10	**NMUBSCITP408**	Mathematics for Information Technology II Practical		1

Total credits		**22**

Programme : B. Sc. IT (Information Technology)					**Semester : IV**		
Course : Core Java					**Code : NMUBSCIT401**		
Suggested Lectures per week				**04**			
Practical Session per week (per Batch)				**01**			
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Component 1**	**Component 2**
60	**15 X 2 = 30**	**Nil**	**04+01**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**
Internal Component (Theory Break up)							
Class Test Duration 20 Mins				**Assignments**			
10 Marks				**15 Marks**			
Internal Component (Practical Break up)							
Examination (Duration 1 ½ Hrs)				**Mini Project/Case study/** **Field Visit (report to be submitted and certified prior to practical examination)**			
30 Marks				20 Marks			
Learning Objectives : 1. Learning Core java programming concepts 2. To learn object oriented concepts with Java 3. To learn multithreading concepts of Java 4. To learn AWT concepts							

.

Learning Outcomes :

1. Proficiency in Core java programming
2. Ability to design object oriented paradigm
3. Ability to write multithreaded programs
4. Ability to design AWT programs

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	**Introduction:** History, architecture and its components, Java Class File, Java Runtime Environment, The Java Virtual Machine, JVM Components, The Java API, java platform, java development kit, Lambda Expressions, Methods References, Type Annotations, Method Parameter Reflection, setting the path environment variable, Java Compiler And Interpreter, java programs, java applications, main(), public, static, void, string[] args, statements, white space, case sensitivity, identifiers, keywords, comments, braces and code blocks, variables, variable name **Data types:** primitive data types, Object Reference Types, Strings, Auto boxing, operators and properties of operators, Arithmetic operators, assignment operators, increment and decrement operator, relational operator, logical operator, bitwise operator, conditional operator.	Classroom learning Presentation	12

II	**Control Flow Statements:** The If…Else If…Else Statement, The Switch…Case Statement **Iterations:** The While Loop, The Do … While Loop, The For Loop, The Foreach Loop, Labeled Statements, The Break And Continue Statements, The Return Statement **Classes:** Types of Classes, Scope Rules, Access Modifier, Instantiating Objects from A Class, Initializing The Class Object And Its Attributes, Class Methods, Accessing A Method, Method Returning a Value, Method's Arguments, Method Overloading, Variable Arguments [Varargs], Constructors, this Instance, super Instance, Characteristics of Members of a Class, constants, this instance, static fields of a class,static methods of a class, garbage collection.	Classroom learning Presentation	12
III	**Inheritance:** Derived Class Objects, Types of Inheritance and Access Control, Default Base Class Constructors, this and super keywords. Abstract Classes and Interfaces, Abstract Classes, Abstract Methods, Interfaces, What Is an Interface? How Is an Interface Different from an Abstract Class? Multiple Inheritance, Default Implementation,	Classroom learning Presentation	12
	Adding New Functionality, Method Implementation, Classes V/s Interfaces, defining an Interface, Implementing Interfaces. **Packages:** Creating Packages, Default Package, Importing Packages, Using A Package.		
IV	**Enumerations, Arrays:** Two Dimensional Arrays, Multi-Dimensional Arrays, Vectors, Adding Elements to A Vector, Accessing Vector Elements, searching for Elements in A Vector, Working with The Size of the Vector. Methods of the Vector class **Multithreading:** the thread control methods, thread life cycle, the main thread, creating a thread, extending the thread class. **Exceptions:** Catching Java Exceptions, Catching Run-Time Exceptions, Handling Multiple Exceptions, The finally Clause, The throws Clause **Byte streams:** reading console input, writing console output, reading file, writing file, writing binary data, reading binary data, getting started with character streams, writing file, reading file	Classroom learning Presentation.	12

V	**Event Handling:** Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes and inner classes. **Abstract Window Toolkit:** Window Fundamentals, Component, Container, Panel, Window, Frame, Canvas. Components – Labels, Buttons, Check Boxes, Radio Buttons, Choice Menus, Text Fields, Text, Scrolling List, Scrollbars, Panels, Frames **Applets :**What are applets in java?,Applet life cycle,init(),paint() methods in applet class. Applet programming. **Layouts:** Flow Layout, Grid Layout, Border Layout, Card Layout.	Classroom learning Presentati on	12

List of Practical Questions:

1. Java Basics
   a. Write a Java program that takes a number as input and prints its multiplication table up to 10.
   b. Write a Java program to display the following pattern.
   ```
   *****
    ****
     ***
     **
      *
   ```
   c. Write a Java program to print the area and perimeter of a circle.

2. Use of Operators
   a. Write a Java program to add two binary numbers.
   b. Write a Java program to convert a decimal number to binary number and vice versa.
   c. Write a Java program to reverse a string.

3. Java Data Types
   a. Write a Java program to count the letters, spaces, numbers and other characters of an input string.
   b. Implement a Java function that calculates the sum of digits for a given char array consisting of the digits '0' to '9'. The function should return the digit sum as a long value.
   c. Find the smallest and largest element from the array

4. Methods and Constructors
   a. Designed a class SortData that contains the method asec () and desc ().

b. Designed a class that demonstrates the use of constructor and destructor.

c. Write a java program to demonstrate the implementation of abstract class.

5. Inheritance

a. Write a java program to implement single level inheritance.

b. Write a java program to implement method overriding

c. Write a java program to implement multiple inheritance.

6. Packages and Arrays

a. Create a package, Add the necessary classes and import the package in java class.

b. Write a java program to add two matrices and print the resultant matrix.

c. Write a java program for multiplying two matrices and print the product for the same.

7. Vectors and Multithreading

a. Write a java program to implement the vectors.

b. Write a java program to implement thread life cycle.

c. Write a java program to implement multithreading.

8. File Handling

a. Write a java program to open a file and display the contents in the console window.

b. Write a java program to copy the contents from one file to other file.

c. Write a java program to read the student data from user and store it in the file.

9. GUI and Exception Handling

a. Design a AWT program to print the factorial for an input value.

b. Design an AWT program to perform various string operations like reverse string, string concatenation etc.

c. Write a java program to implement exception handling.

10. GUI Programming.

a. Design an AWT application that contains the interface to add student information and display the same.

b. Design a calculator based on AWT application.

c. Design an AWT application to generate result marks sheet.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year

1	Programming with JAVA A Primer	E. Balagurusamy	McGraw Hill	6th	2018
2.	Core Java 8 for Beginners	Vaishali Shah, Sharnam Shah	SPD	1st	2015
3.	Java: The Complete Reference	Herbert Schildt	McGraw Hill	9th	2014
4.	Murach’s beginning Java with Net Beans	Joel Murach , Michael Urban	SPD	1st	2016
5.	Core Java, Volume I: Fundamentals	Hortsman	Pearson	9th	2013
6.	Core Java, Volume II: Advanced Features	Gary Cornell and Hortsman	Pearson	8th	2008
7.	Core Java: An Integrated Approach	R. Nageswara Rao	DreamTech	1st	2008

Programme : B. Sc. IT (Information Technology)	**Semester : III**
Course : Embedded System	**Code : NMUBSCIT406**
Suggested Lectures per week	**04**
Practical Session per week (per Batch)	**01**

Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Component 1**	**Component 2**
60	**15 X 2 = 30**	**Nil**	**04+01 = 05**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**

Internal Component (Theory Break up)	
Class Test Duration 20 Mins	**Assignments**
10 Marks	**15 Marks**

Internal Component (Practical Break up)	
Examination (Duration 1 ½ Hrs)	**Mini Project/Case study/ Field Visit (report to be submitted and certified prior to practical examination)**
30 Marks	20 Marks

Learning Objectives :

1. To understand concept of embedded systems and their applications in real world
2. To learn programming basics for embedded systems
3. To understand structure of 8051 and PIC microcontroller basics
4. To learn design and development of RTOS

Learning Outcomes :

1. Ability to operate, design and programme embedded system
2. Ability to design mini applications and projects using embedded system
3. Capacity in use, design and developing RTOS

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	**Introduction**: Embedded Systems and general purpose computer systems, history, classifications, applications and purpose of embedded systems **Core of embedded systems**: microprocessors and microcontrollers, RISC and CISC controllers, Big endian and Little-endian processors, Application specific ICs, Programmable logic devices, COTS, Memory organization, sensors and actuators, communication interface, embedded firmware, other system components.	Classroom learning Presentations Simulations	12
II	**Characteristics and quality attributes of embedded systems:** Characteristics, operational and non-operational quality attributes. **Embedded Systems Application and Domain Specific:** Application specific – washing machine, domain specific - automotive. **Programming embedded systems**: structure of embedded program, infinite loop, compiling, linking and debugging	Classroom learning Presentations Simulations	12

III	**8051 Microcontroller** Microcontrollers and Embedded processors, Overview of 8051 family. 8051 Microcontroller hardware, Input/output pins, Ports, and Circuits, External Memory. **8051 Programming in C:** Data Types and time delay in 8051 C, I/O Programming, Logic operations, Data conversion Programs., Timers, Interfacing 7 segment display, Interfacing LCD display	Classroom learning Presentations Simulations	12
IV	**PIC Microcontroller** Introduction to PIC, Architecture of PIC microcontroller, Pin Diagram RISC architecture and its features, **PIC programming in C** Data Types and time delay in PIC C, I/O Programming, Logic operations, Data conversion Programs	Classroom learning Presentations Simulations	12
V	**Real Time Operating System (RTOS):** Operating system basics, types of operating systems **Design and Development:** Embedded system development Environment – IDE, types of file generated on cross compilation, disassembler/ de-compiler, simulator, emulator and debugging, Trends in embedded industry	Classroom learning Presentations Simulations	12

List of Practical Questions:

1. Design and develop a reprogrammable embedded computer using 8051microcontrollers and to show the following aspects.

    a. Programming

    b. Execution

    c. Debugging

2.

    A Configure timer control registers of 8051 and develop a program to generategiven time delay.

    B To demonstrate use of general purpose port i.e. Input/ output port of two controllers for data transfer between them.

3.

    A Port I / O: Use one of the four ports of 8051 for O/P interfaced to eight LED's. Simulate binary counter (8 bit) on LED's

    B To interface 8 LEDs at Input-output port and create different patterns.

    C To demonstrate timer working in timer mode and blink LED without usingany loop delay routine.

4.

A Serial I / O: Configure 8051 serial port for asynchronous serial communication with serial port of PC exchange text messages to PC and display on PC screen. Signify end of message by carriage return.

B To demonstrate interfacing of seven-segment LED display and generate counting from 0 to 99 with fixed time delay.

C Interface 8051 with D/A converter and generate square wave of given frequency on oscilloscope.

5.

A Interface 8051 with D/A converter and generate triangular wave of given frequency on oscilloscope.

B Using D/A converter generate sine wave on oscilloscope with the help of lookup table stored in data area of 8051.

6. Interface stepper motor with 8051 and write a program to move the motor through a given angle in clock wise or counter clock wise direction.

7. Generate traffic signal.

8. Implement Temperature controller.

9. Implement Elevator control.

10. Interface LCD in 8051

11. Using FlashMagic

A To demonstrate the procedure for flash programming for reprogrammable embedded system board using FlashMagic

B To demonstrate the procedure and connections for multiple controllers programming of same type of controller with same source code in one go, using flash magic.

Books and References:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Programming Embedded Systems in C and C++	MichaelBarr	O'Reilly	First	1999
2.	Introduction to embedded systems	Shibu K V	Tata Mcgraw-Hill	First	2012

3.	The 8051 Microcontroller and Embedded Systems	Muhammad Ali Mazidi	Pearson	Second	2011
4.	Embedded Systems	Rajkamal	Tata Mcgraw-Hill		
5.	PIC Microcontroller and Embedded Systems Using ASM &C for PIC18	Mazidi	Pearson		

Programme : B. Sc. IT (Information Technology)	**Semester : IV**
Course : Mobile Programming	**Code : NMUBSCIT407**
Suggested Lectures per week	**03**
Practical Session per week (per Batch)	**01**
Teaching Scheme	**Evaluation Scheme**

Lecture	**Practical**	**Tutorial**	**Credits**	**Theory**		**Practical**	
				Internal	**External**	**Internal**	**External**
15X3= 45	**15 X 2 = 30**	**Nil**	**03+01=04**	**25 Marks**	**75 Marks**	**20 Marks**	**30 Marks**

Internal Component (Theory Break up)	
Class Test Duration 30 Mins	**Assignments**
15 Marks	**10 Marks**

Internal Component (Practical Break up)	
Examination (Duration 1 ½ Hrs)	**Mini Project/Case study/ Field Visit (report to be submitted and certified prior to practical examination)**
30 Marks	20 Marks

Learning Objectives :

1. To understand various types of mobile apps and need of Angular JS
2. To learn about core component of Iconic
3. To understand side menus, modals, action sheets, and ionScroll 126
4. To learn various techniques of professional app development

Learning Outcomes :

1. Ability to program AngularJS
2. Ability to develop wheather application, bitcoin application
3. Ability to program modern gesture based events in mobile app

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	**Introducing Ionic and hybrid apps** What is Ionic? Types of mobile experiences, Understanding how the Ionic stack works, Why Ionic? , Prerequisites for building apps with Ionic Supported mobile devices and platforms **What you need to know about AngularJS** AngularJS at a glance, Setting up for the chapter project , Basics for an Angular app, Controllers: for controlling data and business logic, Loading data: using the controller to load and display data in the view, Handling click events to select a note, Create a directive to parse a note with Markdown, Using models to manage contentediting, Saving and deleting a note , Continuing with Angular	Simulation Presentation Classroom Talk	09

II	**Ionic navigation and core components** Set up chapter project, Setting up the app ,Building the home view Using a controller and model for the reservation view , Loading data into the weather view, Infinite scroll with cards for the restaurants view, Using the slidebox component for app intro tour **Tabs, advanced lists, and form components** Setup chapter project, ionTabs: adding tabs and navigation, Adding ionNavView for each tab, Loading and displaying current Bitcoin rates, Display a currency’s details in the same tab view , Refresh the Bitcoin rates and display help ,Charting historical data ,Currencies tab with list reorderingand toggles	Simulation Presentation Classroom Talk	09
III	**Weather app, using side menus, modals, action sheets,** **and ionScroll 126** Setting up the chapter project, Setting up the side menu and views Searching for locations, Adding settings view and data services Setting up the weather view ,ionScroll: building custom scrolling content, Action sheet: displayinga list of options , ionModal: displaying the sunrise and sunset chart, Popup: alert and confirm changes to	Simulation Presentation Classroom Talk	09
IV	**Advanced techniques for professional apps** Set up chapter project, Custom Ionic styling using Sass , How to support online and offline mode ,Handling gesture events in Ionic Storing data for persistence, Building one app for multiple platforms Modify default behaviors with $ionicConfigProvider **Building and publishing apps** Building for production: an overview, Building icons and splash-screen assets, Preparing your app for production, Building Android apps and publishing to Google Play, Building iOS apps and publishing to the AppStore	Simulation Presentation Classroom Talk	09

V	**What is React Native?** Advantages of React Native, Risks and Drawbacks **Working with React Native**- How Does React Native Work?, Rendering Lifecycle, Creating Components in React Native, Host Platform APIs **Building Your First Application** Setting Up Your Environment, Creating a New Application, Exploring the Sample Code, Buildinga Weather App	Simulation Presentation Classroom Talk	09

<u>List of Practical Questions:</u>

The practical's will be based on HTML5, CSS, CORDOVA and PhoneGAP API.

Setting up Ionic, React Native Project and environment.

1.

a. Creating and building simple "Hello World" App using Ionic

b. Adding and Using Buttons

c. Adding and Using Event Listeners

2.

a. Creating and Using Functions

b. Using Events

c. Handling and Using Back Button

3.

a. Installing and Using Plugins

b. Installing and Using Battery Plugin

c. Installing and Using Camera Plugin

4.

a. Installing and Using Contacts Plugin

b. Installing and Using Device Plugin

c. Installing and Using Accelerometer Plugin

5.

a. Install and Using Device Orientation plugin

b. Install and Using Device Orientation plugin

c. Create and Using Prompt Function

6.

a. Installing and Using File Plugin

b. Installing and Using File Transfer Plugin

c. Using Download and Upload functions

7.

a. Installing and Using Globalization Plugin
b. Installing and Using Media Plugin
c. Installing and Using Media Capture Plugin

8.

a. Installing and Using Network Information Plugin
b. Installing and Using Splash Screen Plugin
c. Installing and Using Vibration Plugin

9.

a. Developing Single Page Apps
b. Developing Multipage Apps
c. Storing Data Locally in a Cordova App

10.

a. Use of sqlite plugin with Ionic/React Native
b. Using Sqlite read/write and search

Books and References:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Ionic in Action Hybrid Mobile Apps With Ionic And Angularjs	Jeremy Wilken	Manning Publications Co.		2016
2.	Learning React Native Building Mobile Applications with JavaScript	Bonnie Eisenman	O’Reilly Media, Inc.,		2016

Programme : B. Sc. IT (Information Technology)					Semester : IV		
Course : Software Engineering					Code : NMUBSCIT404		
Suggested Lectures per week				04			
Practical Session per week (per Batch)				01			
Teaching Scheme				Evaluation Scheme			
Lecture	Practical	Tutorial	Credits	Theory		Practical	
				Internal	External	Component 1	Component 2
60	15 X 2 = 30	Nil	03+01 = 4	25 Marks	75 Marks	20 Marks	30 Marks
Internal Component (Theory Break up)							
Class Test Duration 20 Mins				Assignments			
10 Marks				15 Marks			
Internal Component (Practical Break up)							
Examination (Duration 1 ½ Hrs)				Mini Project/Case study/ Field Visit (report to be submitted and certified prior to practical examination)			
30 Marks				20 Marks			
Learning Objectives :							

1. To provide understanding the concepts of Software engineering and Project Management
2. To identify the various approaches to project development, cost estimation 3. To learn effort estimation tools and use them

4. To learn about advances in software engineering – Service based, cloud based Software Engineering

Learning Outcomes :

1. Ability to decide a suitable model for projects
2. Ability to distinguish software engineering, process management and software project management
3. To be aware of advanced technologies with respect to SaaS, Cloud basedSoftware engineering requirement.

Pedagogy : Classroom learning , Presentation.

I	**Introduction:** What is software engineering? Software Development Life Cycle, Requirements Analysis, Software Design, Coding, Testing,Maintenance etc. **Software Requirements:** Functional and Non functional requirements, User Requirements, System Requirements, Interface Specification, Documentation of the software requirements. **Software Processes:** Process and Project, Component Software Processes. **Software Development Process Models.** • Waterfall Model. • Prototyping. • Iterative Development. • The RAD Model • Agile Model. **System Models:** Models and its types, Context Models, Behavioural Models, Data Models, ObjectModels, Structured Methods. **Architectural Design:** Architectural Design Decisions, System Organization, Modular Decomposition Styles, Control Styles, Reference Architectures. **User Interface Design:** Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation.	Classroom learning	12

II	**Software Project Management** Introduction to Software Project Management: Introduction, Why is Software Project Management Important? What is a Project? Software Projects versus Other Types of Project, Contract Management and Technical Project Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some Ways of Categorizing Software Projects,Project Charter, Stakeholders, Setting Objectives, The Business Case, Project Success and Failure, What is Management? Management Control, Project Management Life Cycle, Traditional versus Modern Project Management Practices.	Classroo m learning.	12
III	**Software Effort Estimation**: Introduction, Where are the Estimates Done? Problems with Over- and UnderEstimates, The Basis for Software Estimating, Software Effort Estimation Techniques, Bottom- up Estimating, The Top-down Approach and Parametric Models, Expert Judgment, Estimating by Analogy, Albrecht Function Point Analysis, Function Points Mark II, COSMIC Full Function Points, COCOMO II: A Parametric Productivity Model, Cost Estimation, Staffing Pattern, Effect of Schedule Compression, Capers Jones Estimating Rules of Thumb	Classroom learning.	12
IV	**An Overview of Project Planning**: Introduction to Step Wise Project Planning, Step 0: Select Project, Step 1: Identify Project Scope and Objectives, Step 2: Identify Project Infrastructure, Step 3: Analyse Project Characteristics, Step 4: Identify Project Products and Activities, Step 5: Estimate Effort for Each Activity, Step 6: Identify Activity Risks, Step 7: Allocate Resources, Step 8: Review/Publicize Plan, Steps 9and 10: Execute Plan/Lower Levels of Planning	Classroom learning.	12

V	**Service Oriented Software Engineering:** Services as reusable components, Service Engineering, Software Development with Services. **Software reuse.** The reuse landscape, Application frameworks, Software product lines, COTS product reuse. **Distributed software engineering**: Distributed systems issues, Client–server computing, Cloud basedsoftware engineering, Architectural patterns for distributed systems, Software as a service	Classroom learning	12

List of Practical

List of Practical Questions: (To be executed using Star UML or any similar software)

1. Study and implementation of class diagrams.
2. Study and implementation of Use Case Diagrams.
3. Study and implementation of Entity Relationship Diagrams.
4. Study and implementation of Sequence Diagrams.
5. Study and implementation of State Transition Diagrams.
6. Study and implementation of Data Flow Diagrams.
7. Study and implementation of Collaboration Diagrams.
8. Study and implementation of Activity Diagrams.
9. Study and implementation of Component Diagrams.
10. Study and implementation of Deployment Diagrams.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Software Engineering, edition,	Ian Somerville	Pearson Education.	Ninth	

2.	Software Engineering	Pankaj Jalote	Narosa Publication		
3.	Software engineering, a practitioner’s approach	Roger Pressman	Tata Mcgrawhill	Seventh	
4.	Software Engineering principles and practice	WS Jawadekar	Tata Mcgrawhill		
5.	Software Engineering A Concise Study	S.A Kelkar	PHI India.		

<table>
<tr><td colspan="5">Programme : B. Sc. IT (Information Technology)</td><td colspan="3">Semester : IV</td></tr>
<tr><td colspan="5">Course : Mathematics for Information Technology II</td><td colspan="3">Code : NMUBSCIT408</td></tr>
<tr><td colspan="4">Suggested Lectures per week</td><td colspan="4">04</td></tr>
<tr><td colspan="4">Practical Session per week (per Batch)</td><td colspan="4">01</td></tr>
<tr><td colspan="4">Teaching Scheme</td><td colspan="4">Evaluation Scheme</td></tr>
<tr><td rowspan="2">Lecture</td><td rowspan="2">Practical</td><td rowspan="2">Tutorial</td><td rowspan="2">Credits</td><td colspan="2">Theory</td><td colspan="2">Practical</td></tr>
<tr><td>Internal</td><td>External</td><td>Component 1</td><td>Component 2</td></tr>
<tr><td>60</td><td>15 X 2 = 30</td><td>Nil</td><td>04+01</td><td>25 Marks</td><td>75 Marks</td><td>20 Marks</td><td>30 Marks</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">Internal Component (Theory Break up)</td></tr>
<tr><td colspan="4">Class Test Duration 20 Mins</td><td colspan="4">Assignments</td></tr>
<tr><td colspan="4">10 Marks</td><td colspan="4">15 Marks</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">Internal Component (Practical Break up)</td></tr>
<tr><td colspan="4">Examination (Duration 1 ½ Hrs)</td><td colspan="4">Mini Project/Case study/
Field Visit (report to be submitted and certified prior to practical examination)</td></tr>
<tr><td colspan="4">30 Marks</td><td colspan="4">20 Marks</td></tr>
<tr><td colspan="8"></td></tr>
</table>

Learning Objectives :

1. To learn basics of data exploration
2. To learn basics of probability theory and statistics
3. To learn sampling estimation and sampling decision
4. To learn curve fitting technique
5. To understand correlation theory

Learning Outcomes :

1. Ability to perform software computations for data exploration
2. Ability to apply probability and sampling theory data
3. Ability to perform Chi-Square test using R

Pedagogy : Classroom learning , Presentation.

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	**Moments, Skewness, and Kurtosis: Moments**, Moments for Grouped Data, Relations Between Moments, Computation of Moments for Grouped Data, Charlie's Check and Sheppard's Corrections, Moments in Dimensionless Form, Skewness, Kurtosis, Population Moments, Skewness, and Kurtosis, Software Computation of Skewness and Kurtosis.	Classroom learning	12

II	**Elementary Probability Theory**: Definitions of Probability, Conditional Probability; Independent and Dependent Events, Mutually Exclusive Events, Probability Distributions, Mathematical Expectation, Relation Between Population, Sample Mean, and Variance, Combinatorial Analysis, Combinations, Stirling’s Approximation to n!,Relation of Probabilityto Point Set Theory, Euler or Venn Diagrams and Probability. **Elementary Sampling Theory:** Sampling Theory, Random Samples and Random Numbers, Sampling with and Without Replacement, Sampling Distributions, Sampling Distribution of Means, Sampling Distribution of Proportions, Sampling Distributions of Differences and Sums, Standard Errors, Software Demonstration of Elementary Sampling Theory.	Classroom learning.	12
III	**Statistical Estimation Theory**: Estimation ofParameters, Unbiased Estimates, Efficient Estimates, Point Estimates and Interval Estimates; Their Reliability, Confidence-Interval Estimates of Population Parameters, Probable Error. **Statistical Decision Theory:** Statistical Decisions, Statistical Hypotheses, Tests of Hypotheses and Significance, or Decision Rules, Type I and Type II Errors, Level of Significance, Tests Involving Normal Distributions, Two-Tailed and One-Tailed Tests, Special Tests, Operating-Characteristic Curves; the		
	Power of a Test, p-Values for Hypotheses Tests, Control Charts, Tests Involving Sample Differences, Tests Involving Binomial Distributions. **Statistics in R:** mean, median, mode, Normal Distribution , Binomial Distribution, Frequency Distribution in R.		

	Small Sampling Theory: Small Samples, Student’s t Distribution, Confidence Intervals, Tests of Hypotheses and Significance, The Chi-Square Distribution, Confidence Intervals for Sigma, Degrees of Freedom, The F Distribution. **The Chi-Square Test**: Observed and Theoretical Frequencies, Definition of chi-square, Significance Tests, The Chi-Square Test for Goodness of Fit, Contingency Tables, Yates’ Correction for Continuity, Simple Formulas for Computing chisquare, Coefficient of Contingency, Correlation of Attributes, Additive Property of chisquare	Classroom learning.	12
IV	**Curve Fitting and the Method of Least Squares:** Relationship Between Variables, Curve Fitting, Equations of Approximating Curves, Freehand Method of Curve Fitting, The Straight Line, The Method of Least Squares, The Least-Squares Line, Nonlinear Relationships, The Least-Squares Parabola, Regression, Applications to Time Series, Problems Involving More Than Two Variables.	Classroom learning.	12
V	**Correlation Theory:** Correlation and Regression, Linear Correlation, Measures of Correlation, The Least-Squares Regression Lines, Standard Error ofEstimate, Explained and Unexplained Variation, Coefficient of Correlation, Remarks Concerning the Correlation Coefficient, Product- Moment Formula for the Linear Correlation Coefficient, Short Computational Formulas, Regression Lines and the Linear Correlation Coefficient, Correlation of Time Series, Correlation of Attributes, Sampling Theory of Correlation, Sampling Theory of Regression	Classroom learning	12

<u>List of Practical Questions:</u>

1. Using R import the data from Excel / .CSV file and draw the skewness.

2. Import the data from Excel / .CSV and perform the hypothetical testing.
3. Import the data from Excel / .CSV and perform the Chi-squared Test.
4. Using R perform the binomial and normal distribution on the data.
5. Perform the Linear Regression using R.
6. Compute the Least squares means using R.
7. Compute the Linear Least Square Regression
8. Compute correlation in R using o Pearson correlation formula
   o Spearman correlation formula o Kendall correlation formula

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	STATISTICS	Murray R. Spiegel, Larry J. Stephens.	McGRAW – HILL ITERNATIONAL	FOURTH	
2.	A Practical Approach using R	R.B. Patil, H.J. Dand and R. Bhavsar	SPD	1st	2017
3.	FUNDAMENTA LOF MATHEMATICAL STATISTICS	S.C. GUPTA and V.K. KAPOOR	SULTAN CHAND and SONS	ELEVENTH REVISED	2011
4.	MATHEMATICAL STATISTICS	J.N. KAPUR and H.C. SAXENA	S. CHAND	TWENTIETH REVISED	2005