

Bachelor of Information Technology

(BSc IT)

3 Years Undergraduate Program

Program Objectives

1. To cater to the needs of enormous IT sector in India as well as abroad.
2. Enable learners to think rationally, productively and critically in developing strong, extensible and highly maintainable technological solutions to simple and complex systems.
3. Enhance learners in their knowledge and skills to be employed and outshine in IT professional careers.
4. Empower learners to adhere to the highest standards of ethics, including relevant industry and organizational codes of conduct.

Program Outcomes

1. Skill Set 1: Software development fundamentals, including programming, data structures, algorithms and complexity
2. Skill Set 2: Systems fundamentals, including architectures and organization, operating systems, networking and communication, parallel and distributed computation, and security
3. Skill Set 3: Mathematics fundamentals, including discrete structures, statistics and calculus
4. Skill Set 4 : Software engineering fundamentals, including software analysis and design, evaluation and testing, and software engineering processes
5. Skill Set 5 : Application fundamentals, including information management and intelligent applications
6. Skill Set 6 : Multiple programming languages, paradigms, and technologies
7. Skill Set 7 : Eligibility for Master's degree (MCA, MBA, MSc IT/CS)

Semester I

Course Code	Course Type	Course Title	Credits
NMUBSCIT101	Core Subject	<u>Structured Programming using C</u>	04
NMUBSCIT102	Core Subject	<u>Web Programming</u>	04
NMUBSCIT103	Core Subject	<u>Basics of Computer Hardware & Operating Systems</u>	04
NMUBSCIT104	Core Subject	<u>Discrete Mathematics</u>	04
NMUBSCIT105	Ability Enhancement Skills	<u>Communication Skills</u>	03
NMUBSCITP101	Course Type Practical	C Programming Practical	01
NMUBSCITP102	Core Subject Practical	Web Programming Practical	01
NMUBSCITP103	Core Subject Practical	Hardware & Operating System Practical	01
NMUBSCITP104	Core Subject Practical	Discrete Mathematics Practical	01
NMUBSCITP105	Ability Enhancement Skills Practical	Communication Skills Practical	01
		Total Credits	24

Semester II

Course Code	Course Type	Course Title	Credits
NMUBSCIT201	Core Subject	<u>Object Oriented Programming</u>	04
NMUBSCIT202	Core Subject	<u>Microprocessor Design and Programming</u>	04
NMUBSCIT203	Core Subject	<u>Notion of Operating System</u>	04
NMUBSCIT204	Core Subject	<u>Statistical Methods for Data Science</u>	04
NMUBSCIT205	Ability Enhancement Skills	<u>Digital Marketing</u>	03
NMUBSCITP201	Course Type Practical	Object oriented Programming Practical	01
NMUBSCITP202	Core Subject Practical	Microprocessor Design and Programming Practical	01
NMUBSCITP203	Core Subject Practical	Linux Operating System Practical	01
NMUBSCITP204	Core Subject Practical	Statistical Methods for Data Science Practical	01
NMUBSCITP205	Ability Enhancement Skills Practical	Digital Marketing Practical	01
		Total Credits	24

SEMESTER

I

Programme : B. Sc IT (Information Technology)				Semester I			
Course : Structured Programming using C				Code : NMUBSCIT101			
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							
<ol style="list-style-type: none"> 1. To understand writing algorithms and programs in programming language 2. To make students develop their logic skills 							
Learning Outcomes							
<ol style="list-style-type: none"> 1. Understanding fundamentals of Algorithm, complexity concept 2. Understanding programming concept of loops and conditions 3. Understanding higher order programming concept 4. Understanding usage of user defined functions 5. Understanding concept of pointers 							
Pedagogy							
Presentations, Programming Simulators							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	<p>Introduction to Algorithms: Fundamentals of algorithms: Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures.</p> <p>Introduction: Types of Programming languages, History, features and application.</p> <p>Fundamentals: Structure of a program. Compilation and Execution of a Program, Character Set, identifiers and keywords, data types, constants, variables and arrays, declarations, expressions, statements, Variable definition, symbolic constants.</p> <p>Data Input and output: Single character input and output, entering input data, scanf function, printf function, gets and puts functions, interactive programming.</p>	Presentations, programming Exercises	12
II	<p>Operators and Expressions: Arithmetic operators, unary operators, relational and logical operators, assignment operators, assignment operators, the conditional operator, library functions. Conditional Statements and Loops: Decision Making Within A Program, Conditions, Relational Operators, Logical Connectives, If Statement, If-Else Statement, Loops: While Loop, Do While, For Loop. Nested Loops, Infinite Loops, Switch Statement</p>	Presentations, programming Exercises	12
III	<p>Program structure: Only Introduction to Storage classes, automatic variables, external variables, static variables, multifile programs, more library functions, Preprocessor: Features, #define and #include, Directives and Macros Arrays: Definition, processing, passing arrays to functions, multidimensional arrays, arrays and strings.</p>	Presentations, programming Exercises	12
IV	<p>Functions: Overview, defining a function, accessing a function, passing arguments to a function, specifying argument data types, function prototypes, recursion, modular programming and functions, standard library of c</p>	Presentations, programming Exercises	12

	functions, prototype of a function, return type, function call, block structure, passing arguments to a function: call by reference, call by value.		
V	<p>Pointers: Fundamentals, declarations, Pointers Address Operators, Pointer Type Declaration, Pointer Assignment, Pointer Initialization, Pointer Arithmetic, Functions and Pointers, Arrays And Pointers, Pointer Arrays, passing functions to other functions</p> <p>Structures and Unions: Structure Variables, Initialization, Structure Assignment, Nested Structure, Structures and Functions, Structures and Arrays: Arrays of Structures, Structures Containing Arrays, Unions, Structures and pointers.</p>	Presentations, programming Exercises	12
	<p>Reference Books</p> <ol style="list-style-type: none"> 1. Programming in ANSI C By E. Balagurusamy Tata McGRAWHill Sixth Edition 2. Programming with C Byron Gottfried Tata McGRAWHill 2nd 1996 3. Programming Logic and Design Joyce Farell Cengage Learning 8th 2014 		
	<p>List of Practical: (Can be done in C language)</p> <p>1. Basic Programs:</p> <ol style="list-style-type: none"> a. Write a program to display the message HELLO WORLD. b. Write a program to declare some variables of type int, float and double. Assign some values to these variables and display these values. c. Write a program to find the addition, subtraction, multiplication and division of two numbers. <p>2. Programs on variables:</p> <ol style="list-style-type: none"> a. Write a program to swap two numbers without using third variable. b. Write a program to find the area of rectangle, square and circle. c. Write a program to find the volume of a cube, sphere, and cylinder. <p>3. Conditional statements and loops(basic)</p> <ol style="list-style-type: none"> a. Write a program to enter a number from the user and display the month name. If number >13 then display invalid input using switch case. b. Write a program to check whether the number is even or odd. 		

	<p>c. Write a program to check whether the number is positive, negative or zero.</p> <p>d. Write a program to find the factorial of a number.</p> <p>e. Write a program to check whether the entered number is prime or not.</p> <p>f. Write a program to find the largest of three numbers.</p> <p>4. Conditional statements and loops(advanced)</p> <p>a. Write a program to find the sum of squares of digits of a number.</p> <p>b. Write a program to reverse the digits of an integer.</p> <p>c. Write a program to find the sum of numbers from 1 to 100.</p> <p>d. Write a programs to print the Fibonacci series.</p> <p>e. Write a program to find the reverse of a number.</p> <p>f. Write a program to find whether a given number is palindrome or not.</p> <p>g. Write a program that solve the quadratic equation</p> <p>h. Write a program to check whether the entered number is Armstrong or not.</p> <p>i. Write a program to count the digit in a number</p> <p>5. Programs on patterns:</p> <p>a. Programs on different patterns.</p> <p>6. Functions:</p> <p>a. Programs on Functions.</p> <p>7. Recursive functions</p> <p>a. Write a program to find the factorial of a number using recursive function.</p> <p>b. Write a program to find the sum of natural number using recursive function.</p> <p>8. Arrays</p> <p>a. Write a program to find the largest value that is stored in the array.</p> <p>b. Write a program using pointers to compute the sum of all elements stored in an array.</p> <p>c. Write a program to arrange the 'n' numbers stored in the array in ascending and descending order.</p> <p>d. Write a program that performs addition and subtraction of matrices.</p> <p>e. Write a program that performs multiplication of matrices.</p> <p>9. Pointers</p> <p>a. Write a program to demonstrate the use of pointers.</p> <p>b. Write a program to perform addition and subtraction of two pointer variables.</p> <p>10. Structures and Unions</p> <p>a. Programs on structures.</p> <p>b. Programs on unions.</p>		
--	---	--	--

Programme : B. Sc IT (Information Technology)				Semester : I			
Course : Web Programming				Code : NMUBSCIT102			
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	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)				Static Website			
30 Marks				20 Marks			
Learning Objectives :							
To provide insight into emerging technologies to design and develop state of - the art web applications using client-side scripting, server-side scripting, and database connectivity.							
Learning Outcomes :							
<ol style="list-style-type: none"> 1. Designing valid, well-formed, scalable, and meaningful pages using emerging technologies. 2. Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites 3. Development and implementation of client-side scripting language programs. 4. Developing event driven web model 5. Understand practical current trends in web development 							
Pedagogy : Classroom learning , Presentation.							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	Internet and WWW: What is Internet?, Introduction to internet and its applications, Email, telnet, FTP, e-commerce, video conferencing, e-business. Internet service providers, domain name server, internet address World Wide Web (WWW) : World Wide Web and its evolution, uniform resource locator (URL), browsers - internet explorer, netscape navigator, opera, firefox, chrome, mozilla. Search engine, web saver - apache, IIS, proxy server, HTTP protocol.	Presentation	12
II	HTML5: Fundamental Elements of HTML, Formatting Text in HTML, Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page CSS: Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element	Presentation , Programs	12
III	Java Script : Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security. Operators : Assignment Operators, Comparison Operators, Arithmetic Operators, % (Modulus), ++(Increment), --(Decrement), -(Unary Negation), Logical Operators, Short- Circuit Evaluation, String Operators, Special Operators, ? (Conditional operator), ,(Comma operator), delete, new, this, void Statements : Break, comment, continue, delete, do ... while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with, Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math, Number, Object, String, RegExp Document and its associated objects : document, Link, Area, Anchor, Image, Applet, Layer	Presentation Programs	12
IV	Events and Event Handlers : General Information about Events, Defining Event Handlers, event,	Presentation, Programs	12

	onAbort, onBlur, onChange, onClick, onDbClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload.		
V	Current Industry Trends Window based/Linux based architecture. Wordpress Development. Hosting Services, Types of Hosting. CPanel Essentials. Free hosting.	Presentation, Programs	12
	<u>Books and References :</u> <ol style="list-style-type: none"> 1. Thomas Powel , “Web design the complete reference ” ,Tata McGraw Hil. 2. Faithe Wempen,”HTML5 step by step”, Microsoft Press -2011 3. Ivan Bayross, Sharanam Shah ,”PHP 5.1 for Beginners”, SPD - 2013 4. Sharanam Shah ,Vaishali Shah ,” PHP Project for Beginners” ,SPD - 2015 5. JavaScript 2.0 : The Complete Reference, Second Edition by Thomas Powell and Fritz Schneider(2nd Edition) 		
	List of Practical 1. Use of Basic Tags a. Design a web page using different text formatting tags. b. Design a web page with links to different pages and allow navigation between web pages. c. Design a web page demonstrating all Style sheet types 2. Image maps, Tables, Forms and Media a. Design a web page with Imagemaps. b. Design a web page demonstrating different semantics c. Design a web page with different tables. Design a webpages using table so that the content appears well placed. d. Design a web page with a form that uses all types of controls. e. Design a web page embedding with multimedia features. 3. Java Script a. Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series. b. Design a form and validate all the controls placed on the form using Java Script.		

	<p>c. Write a JavaScript program to display all the prime numbers between 1 and 100.</p> <p>a. Write a JavaScript program to accept a number from the user and display the sum of its digits.</p> <p>d. Write a program in JavaScript to accept a sentence from the user and display the number of words in it. (Do not use split () function).</p> <p>e. Write a java script program to design simple calculator.</p> <p>4. Control and looping statements and Java Script references</p> <p>a. Design a web page demonstrating different conditional statements.</p> <p>b. Design a web page demonstrating different looping statements.</p> <p>c. Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp).</p> <p>5. Javascript Practical</p>		
--	---	--	--

Programme : B. Sc IT (Information Technology)				Semester : I			
Course : Basics of Computer Hardware & Operating Systems				Code : NMUBSCIT103			
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 :-							
Students will try to learn:							
<ol style="list-style-type: none"> 1. The concept of various components. 2. The concepts that underpin the disciplines of Analog and digital electronic logic circuits. 3. Various Number system and Boolean algebra. 4. Design and implementation of combinational circuits 5. Design and implementation of Sequential circuits 6. Hardware description language 							
Learning Outcomes :-							
At the end of the course student should be able-							
<ol style="list-style-type: none"> 1. To understand different number systems and their conversions. 2. To analyze and minimize Boolean expressions. 3. To design and analyze combinational circuits. 4. To design and analyze sequential circuits 5. To understand the basic concepts of Operating System 							
Pedagogy : Student-centered pedagogy, Audio Visual Technology(Powerpoint Presentation)							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	<p>Introduction to Electronics: Analog Vs Digital, Introduction to semi-conductors, Semiconductor diodes. Introduction to Transistors.</p> <p>Number Systems and Codes Introduction to number system and conversions: Binary, Octal, Decimal and Hexadecimal Number Systems, Binary arithmetic: addition, subtraction (1's and 2's complement), multiplication and division. Octal and Hexadecimal arithmetic: Addition and Subtraction (7's and 8's complement method for octal) and (15's and 16's complement method for Hexadecimal). Codes: Gray Code, BCD Code, Excess-3 code, ASCII Code. Error Detection and Correction: Hamming codes.</p>	Student-centered pedagogy	12
II	<p>Boolean Algebra and Logic gates: Theorems and Properties of Boolean Algebra, Boolean functions, Boolean function reduction using Boolean laws, Canonical forms, Standard SOP and POS form.</p> <p>Basic Digital gates: NOT , AND , OR , NAND , NOR , EXOR , EX-NOR, positive and negative logic, K-map method 2 variable, 3 variable, 4 variable, Don't care condition, NAND-NOR Realization (Implementation of other gates using universal gates).</p>	Student-centered pedagogy	12
III	<p>Combinational Logic Design : Introduction, Half and Full Adder, Half and Full Subtractor, Four Bit Binary Adder, One digit BCD Adder, code conversion, Encoder and Decoder ,Multiplexers and Demultiplexers</p>	Student-centered pedagogy	12
IV	<p>Sequential Logic Design :</p> <p>Flip Flops : SR, JK, D, T, master slave flip flop, Truth Table, excitation table and conversion</p> <p>Register: Shift register, SISO, SIPO, PISO, PIPO, Bi-directional and universal shift register.</p> <p>Counters: Design of synchronous and asynchronous ,Modulo Counter, Up Down counter IC 74193, Ring and Johnson Counter.</p>	Student-centered pedagogy and Powerpoint Presentation.	12
V	<p>Introduction: What is an operating system? History of operating system, computer hardware, different operating</p>	Case Study	12

	<p>systems, operating system concepts, system calls, operating system structure.</p> <p>Case Study on Windows, LINUX and ANDROID</p> <p>History of Windows OS. Versions of Windows, windows NT, Versions based on windows NT, Windows server</p> <p>Tasks of Windows : Central tasks of operating systems. Introduction to Windows API.</p> <p>What is Linux : History, User interface, Does Linux have future ? Properties of Linux, Linux Flavors</p> <p>History of Android OS. Different versions available. List of features available in Android.</p>		
	<p>Books and References:</p> <ol style="list-style-type: none"> 1. R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill. 2. Bernard Grob, Basic Electronics, 4th Edition 3. N.G.Palan, “Digital Electronics and Logic Design”, Technova 4. M. Morris Mano, “Digital Logic and computer Design”, PHI 5. Donald p Leach, Albert Paul Malvino, “Digital principles and Applications”, Tata McGraw Hill 6. Modern Operating Systems , by Andrew S. Tanenbaum 7. Introduction to Linux: A Hands on Guide by Machtelt Garrels Publication date 20080606 Copyright © 2002, 2003, 2004, 2005, 2006, 2007, 2008 Machtelt Garrels 8. An Introduction to Windows operating system 2nd Edition @2017 , Einar Krogh 9. Web references for case studies : https://www.operating-system.org/betriebssystem/ english/bs-android.htm 		
	<p>List of Practical</p> <p>1. Study of Logic gates and their ICs and universal gates:</p> <ol style="list-style-type: none"> a. Study of AND, OR, NOT, XOR, XNOR, NAND and NOR gates b. IC 7400, 7402, 7404, 7408, 7432, 7486, 74266 c. Implement AND, OR, NOT, XOR, XNOR using NAND gates. d. Implement AND, OR, NOT, XOR, XNOR using NOR gates. <p>2. Implement the given Boolean expressions using minimum number of gates.</p> <ol style="list-style-type: none"> a. Verifying De Morgan’s laws. 		

	<p>b. Implement other given expressions using minimum number of gates.</p> <p>c. Implement other given expressions using minimum number of ICs.</p> <p>3. Implement combinational circuits.</p> <p>a. Design and implement combinational circuit based on the problem given and minimizing using K-maps.</p> <p>4. Implement code converters.</p> <p>a. Design and implement Binary – to – Gray code converter.</p> <p>b. Design and implement Gray – to – Binary code converter.</p> <p>c. Design and implement Binary – to – BCD code converter</p> <p>d. Design and implement Binary – to – XS-3 code converter</p> <p>5. Implement Adder and Subtractor Arithmetic circuits.</p> <p>a. Design and implement Half adder and Full adder.</p> <p>b. Design and implement BCD adder.</p> <p>c. Design and implement XS – 3 adder.</p> <p>d. Design and implement binary subtractor.</p> <p>e. Design and implement BCD subtractor.</p> <p>f. Design and implement XS – 3 subtractor.</p> <p>6. Implement Arithmetic circuits.</p> <p>a. Design and implement a 2-bit by 2-bit multiplier.</p> <p>b. Design and implement a 2-bit comparator.</p> <p>7. Implement Encode and Decoder and Multiplexer and Demultiplexers.</p> <p>a. Design and implement 8:3 encoder.</p> <p>b. Design and implement 3:8 decoder.</p> <p>c. Design and implement 4:1 multiplexer. Study of IC 74153, 74157</p> <p>d. Design and implement 1:4 demultiplexer. Study of IC 74139</p> <p>e. Implement the given expression using IC 74151 8:1 multiplexer.</p> <p>f. Implement the given expression using IC 74138 3:8 decoder.</p> <p>8. Study of flip-flops and counters.</p> <p>a. Study of IC 7473.</p> <p>b. Study of IC 7474.</p> <p>c. Study of IC 7476.</p> <p>d. Conversion of Flip-flops.</p> <p>e. Design of 3-bit synchronous counter using 7473 and required gates.</p> <p>f. Design of 3-bit ripple counter using IC 7473.</p> <p>9. Case Study based on Windows/Linux/Android</p>		
--	---	--	--

Programme : FY B. Sc IT (Information Technology)				Semester - I			
Course : Discrete Mathematics				Code : NMUBSCIT104			
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	4+1	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							
<p>The main objective is to allow the students</p> <ul style="list-style-type: none"> To think analytically, creatively and critically in developing robust, extensible and highly maintainable technological solutions to simple and complex problems. To be capable of managing complex IT projects with consideration of the human, financial and environmental factors. To develop an aptitude to engage in continuing professional development to think analytically, creatively and critically in developing robust, extensible and highly maintainable technological solutions to simple and complex problems. 							
Learning Outcomes							
<ul style="list-style-type: none"> Mathematics Basis of various algorithms and logic building Mathematics basis for recursive functions Mathematics behind sequence and series Mathematics of Graph and Trees Combination and probability concepts. 							
Pedagogy							
PPT + Blackboard.							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	<p>Introduction: Variables, The Language of Sets, The Language of Relations and Function Set Theory: Definitions and the Element Method of Proof, Properties of Sets, Disproofs, Algebraic Proofs, Boolean Algebras, Russel 's12 Paradox and the Halting Problem.</p> <p>The Logic of Compound Statements: Logical Form and Logical Equivalence, Conditional Statements, Valid and Invalid Arguments Quantified Statements: Predicates and Quantified Statements, Statements with Multiple Quantifiers, Arguments with Quantified Statements</p> <p>Elementary Number Theory and Methods of Proof: Introduction to Direct Proofs</p>	PPT +Blackboard	12
II	<p>Rational Numbers, Divisibility, Division into Cases and the Quotient-Remainder Theorem, Floor and Ceiling, Indirect Argument: Contradiction and Contraposition, Two Classical Theorems, Applications in algorithms.</p> <p>Sequences, Mathematical Induction, and Recursion: Sequences, Mathematical Induction, Strong Mathematical Induction and the Well- Ordering Principle for the Integers, Correctness of algorithms.</p>	PPT +Blackboard	12
III	<p>Defining sequences recursively, solving recurrence relations by iteration, Second order linear homogenous recurrence relations with constant coefficients. General recursive definitions and structural induction. Functions: Functions Defined on General Sets, One-to-One and Onto, Inverse Functions, Composition of Functions, Cardinality with Applications to Computability</p> <p>Relations: Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations, Partial Order Relations</p>	PPT +Blackboard	12
IV	<p>Graphs and Trees: Definitions and Basic Properties, Trails, Paths, and Circuits, Matrix Representations of Graphs, Isomorphism's of Graphs, Trees, Rooted Trees, Isomorphism's of Graphs, Spanning trees and shortest paths.</p> <p>Counting and Probability: Introduction, Possibility Trees and the Multiplication Rule, Possibility Trees and the Multiplication Rule</p>	PPT +Blackboard	12
V	<p>Counting Elements of Disjoint Sets: The Addition Rule, The Pigeonhole Principle, Counting Subsets of a Set: Combinations,</p>		

	Combinations with Repetition Allowed, Probability Axioms and Expected Value, Conditional Probability, Bayes' Formula, and Independent Events. Introduction to Game theory.	PPT +Blackboard	12
	Reference <ul style="list-style-type: none"> • Discrete Mathematics with Applications Sussana S. Epp Cengage Learning 4th 2010 • Discrete Mathematics, Schaum's Outlines Series Seymour Lipschutz, Marc Lipson Tata MCGraw Hill 2007 		
	List of Practical <ol style="list-style-type: none"> 1. Set Theory <ol style="list-style-type: none"> a. Inclusion Exclusion principle. b. Power Sets c. Mathematical Induction 2. Probability Theory <ol style="list-style-type: none"> a. Sample space and events b. Finite probability spaces c. Equiprobable spaces d. Addition Principle e. Conditional Probability f. Multiplication theorem for conditional probability g. Independent events h. Repeated trials with two outcomes 3. Counting <ol style="list-style-type: none"> a. Sum rule principle b. Product rule principle c. Factorial d. Binomial coefficients e. Permutations f. Permutations with repetitions g. Combinations h. Combinations with repetitions i. Ordered/Unordered partitions 4. Graph Theory <ol style="list-style-type: none"> a. Paths and connectivity b. Minimum spanning tree c. Isomorphic 		

Programme : B. Sc IT (Information Technology)				Semester : I			
Course : Communication Skills				Code : NMUBSCIT105			
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	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)				Individual Presentations			
30 Marks				20 Marks			
Learning Objectives :							
Possible communications objectives include: <ol style="list-style-type: none"> 1. Creating awareness 2. Imparting knowledge 3. Projecting an image 4. Shaping attitudes 5. Stimulating a want or desire 6. Effecting a sale. 							
Learning Outcomes :							
<ol style="list-style-type: none"> 1. Understanding Business Communications and technology therein 2. Writing professional documents 3. Developing Oral Communication skills for Business 4. Understanding Specific Communication Needs 5. Learning Presentation skills and techniques 							
Pedagogy : Classroom learning , Presentation.							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	Understanding Business Communication: Nature and Scope of Communication, Non-verbal Communication, Cross-cultural communication, Technology-enabled Business Communication. Importance of corporate communication	Presentation	09
II	The Seven Cs of Effective Communication: Completeness, Conciseness, Consideration, Concreteness, Clarity, Courtesy, Correctness. Writing Business Messages and Documents: Business writing, Business Correspondence, Career building and Resume writing. Drafting Business Letters	Presentation, Group Presentation.	09
III	Developing Oral Communication Skills for Business: Effective Listening, Business Presentations and Public Speaking, Conversations, Interviews , Meetings , seminars and Conferences, modern tools used. Group Discussions as selection technique.	Group Discussion.	09
IV	Understanding Specific Communication Needs: Persuasive Strategies in Business Communication, Business Communication Aids. Communication across Functional Areas	Presentation, Group Discussion.	09
V	Presentation Process: Planning the presentations, executing the presentations, Impressing the audience by performing, Planning stage: Brainstorming, mind maps / concept maps, executing stage: chunking theory, creating outlines, Use of templates. Adding graphics to your presentation: Visual communication, Impress stage: use of font, colour, layout, Importance of practice and performance.	Presentation, Group Discussion.	09
	Books and References: 1. Business communication- Edited by Meenakshi Raman and Prakash Singh, Oxford Univeristy Press, 2 nd edition 2. Professional Communication- Aruna Koneru, Tata McGraw Hill		
	List of Practical Questions: 1. Communication Origami, Guessing Game, Guessing the emotion 2. Body Language, Follow All Instructions, Effective Feedback Skills		

	<p>3. The Name Game, Square Talk(Effective Communication), Room 101(Influential and persuasive skills)</p> <p>4. Back to Back Communication, Paper Shapes(Importance of two-way communication), Memory Test(Presentation Skills)</p> <p>5. Exercises on Communication Principles</p> <p>6. Exercises on communication icebreakers</p> <p>7. Communication exercises For the following practicals, Microsoft Office, Open Office, Libre Office or any other software suite can be used.</p> <p>8. Use of word processing tools for communication</p> <p>9. Use of spreadsheet tools for communication</p> <p>10. Use of presentation tools for communication</p>		
--	---	--	--

SEMESTER II

Programme : B. Sc IT (Information Technology)				Semester : II			
Course : Object Oriented Programming				Code : NMUBSCIT201			
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							
<ol style="list-style-type: none"> 1. To understand the concepts of object oriented programming 2. Identify the difference of Object Oriented Programming with Procedure Oriented Programming, 3. Developing programming skills in students. 							
Learning Outcomes							
<ol style="list-style-type: none"> 1. Basic concepts of Object, OOP and classes 2. Constructors and Destructors concept, access levels of objects 3. Inheritance between classes, accessibility therein 4. Polymorphism : Meaning , Applications , Error handling 5. Abstract Classes Concept 							
Pedagogy							
<ol style="list-style-type: none"> 1. Assignments 2. Case studies to be given group wise 3. Presentations 4. Industrial Visits 							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	<p>Object Oriented Methodology: Introduction, Advantages and Disadvantages of Procedure Oriented Languages, what is Object Oriented? What is Object Oriented Development? Object Oriented Themes, Benefits and Application of OOPS.</p> <p>Principles of OOPS: OOPS Paradigm, Basic Concepts of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing</p> <p>Types of operators in C++ : Arithmetic, relational, logical, bitwise, Scope resolution operators in C++, Control Loops :for loop, if loop, if..else loop, While loop,do..while loop, switch.. case statement, conditional operator</p>	<ol style="list-style-type: none"> 1. Assignments 2. Presentation 3. Progression Test 	12
II	<p>Classes and Objects: Simple classes (Class specification, class members accessing), Defining member functions, passing object as an argument, Returning object from functions, friend classes, Pointer to object, Array of pointer to object.</p> <p>Constructors and Destructors: Introduction, Default Constructor, Parameterized Constructor and examples, Destructors</p>	<ol style="list-style-type: none"> 1. Assignments 2. Presentation 3. Progression Test 	12
III	<p>Program development using Inheritance: Introduction, understanding inheritance, Advantages provided by inheritance, choosing the access specifier. Derived class declaration, derived class constructors, class hierarchies, multiple inheritance, multilevel inheritance, containership, hybrid inheritance.</p>	<ol style="list-style-type: none"> 1. Assignments 2. Presentation 3. Progression Test 	12
IV	<p>Polymorphism: Concept of function overloading, overloaded operators, overloading unary and binary operators, overloading comparison operator, overloading arithmetic assignment operator</p> <p>Exception Handling: Introduction, Exception Handling Mechanism, Concept of throw & catch with example</p>	<ol style="list-style-type: none"> 1. Assignments 2. Presentation 3. Progression Test 	12
V	<p>Virtual Functions & Abstract Classes: Introduction to Virtual Functions, Pure Virtual Functions, Abstract classes, Templates: Introduction, Function Template and examples, Class Template and examples.</p>	<ol style="list-style-type: none"> 1. Assignments 2. Presentation 	12
	<p>References:</p> <ol style="list-style-type: none"> 1. Object Oriented Analysis and Design Timothy Budd TMH 3rd 2012 2. Mastering C++ K R Venugopal,Rajkumar Buyya, T Ravishankar Tata McGraw Hill 2nd Edition 2011 		
	<p>List of Practical: To be implemented using object oriented language</p>		

<p>1. Classes and methods</p> <p>a. Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used respectively. Where getInfo() will be private method</p> <p>b. Design the class student containing getData() and displayData() as two of its methods which will be used for reading and displaying the student information respectively. Where getData() will be private method.</p> <p>c. Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not. Where readNo() will be private method.</p> <p>d. Write a program to demonstrate function definition outside class and accessing class members in function definition.</p> <p>2. Using friend functions.</p> <p>a. Write a friend function for adding the two complex numbers, using a single class</p> <p>b. Write a friend function for adding the two different distances and display its sum, using two classes.</p> <p>c. Write a friend function for adding the two matrix from two different classes and display its sum.</p> <p>3. Constructors and method overloading.</p> <p>a. Design a class Complex for adding the two complex numbers and also show the use of constructor.</p> <p>b. Design a class Geometry containing the methods area() and volume() and also overload the area() function .</p> <p>c. Design a class StaticDemo to show the implementation of static variable and static function.</p> <p>4. Operator Overloading</p> <p>a. Overload the operator unary(-) for demonstrating operator overloading.</p> <p>b. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.</p> <p>c. Overload the + for concatenating the two strings. For e.g “Py” + “thon” = Python</p>		
--	--	--

	<p>5. Inheritance</p> <ul style="list-style-type: none"> a. Design a class for single level inheritance using public and private type derivation. b. Design a class for multiple inheritance. c. Implement the hierarchical inheritance. <p>6. Virtual functions and abstract classes</p> <ul style="list-style-type: none"> a. Implement the concept of method overriding. b. Show the use of virtual function c. Show the implementation of abstract class. <p>7. String handling</p> <ul style="list-style-type: none"> a. String operations for string length , string concatenation b. String operations for string reverse, string comparison, c. Console formatting functions. <p>8. Exception handling</p> <ul style="list-style-type: none"> a. Show the implementation of exception handling b. Show the implementation for exception handling for strings c. Show the implementation of exception handling for using the pointers. <p>9. File handling</p> <ul style="list-style-type: none"> a. Design a class FileDemo open a file in read mode and display the total number of words and lines in the file. b. Design a class to handle multiple files and file operations c. Design a editor for appending and editing the files <p>10. Templates</p> <ul style="list-style-type: none"> a. Show the implementation of template class library for swap function. c. Design the template class library for sorting ascending to descending and vice versa 		
--	--	--	--

Programme : B. Sc IT (Information Technology)				Semester : II			
Course : Microprocessor Design and Programming				Code : NMUBSCIT202			
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 :							
<ol style="list-style-type: none"> 1. To Introduce the basics of microprocessors technology and related applications. 2. Study of the architectural details and programming of 16 bit 8086 microprocessor and its interfacing with various peripheral ICs 3. Study of Programming language of 8086 							
Learning Outcomes :							
<ol style="list-style-type: none"> 1. Acquire knowledge about basic microprocessors and introduction to 8086 2. Write programs using 8086 microprocessor 3. Know I/O interfacing and its applications and multiprogramming 4. Know the system bus structure and interfacing of different peripheral devices with 8086 microprocessor 5. Information of memory organization and multiprocessor configurations 							
Pedagogy : Classroom learning , Presentation, Hands on, Simulations							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	<p>Introduction - Overview of Microcomputer Systems, Data Representation, Addresses, General Operation of a Computer, Microprocessors in Digital System Design</p> <p>Microcomputer and Microprocessors The Stored Program Concept, Types of computers, The 80X86 Family of processors</p> <p>8086 Architecture- CPU Architecture, Internal Operation, Machine Language Instructions, Instruction Execution Timing, Introduction to 8088</p>	Presentation	12
II	<p>Assembler Language Programming - Assembler Instruction Format, Data Transfer Instructions, Arithmetic Instructions, Branch Instructions, Loop Instructions, NOP and HLT Instructions, Flag Manipulation Instructions, Logical Instructions, Shift and Rotate Instructions, Directives and Operators, Assembly Process, Translation of Assembler Instructions.</p> <p>Modular Programming - Linking and Relocation, Stacks, Procedures, Interrupts and Interrupt Routines, Macros, Program Design, Program Design Example</p>	Presentation	12
III	<p>Byte and String Manipulation - String Instructions, REP Prefix, Text Editor Example, Table Translation, Number Format Conversions</p> <p>I/O Programming - Fundamental I/O Considerations, Programmed I/O, Interrupt I/O, Block Transfers and DMA, I/O Design Example</p> <p>Introduction to Multiprogramming - Process Management and iRMX, Semaphore Operations, Common Procedure Sharing, Memory Management, Virtual Memory and the 80286</p>	Presentation	12
IV	<p>System Bus Structure - Basic 8086/8088 Configurations, System Bus Timing, Interrupt Priority Management, Bus Standards</p> <p>I/O Interfaces - Serial Communication Interfaces, Parallel Communication, Programmable Timers and Event Counters, Keyboard and Display, DMA Controllers,</p>	Presentation	12

	Diskette Controllers, Maximum Mode and 16-Bit Bus Interface Designs		
V	Semiconductor Memory - General Memory Organization, Static RAM Devices, Dynamic RAM Devices, Backup Power for Semiconductor Memories, ROM Devices Multiprocessor Configurations - Queue Status and the LOCK Facility, 8086/8088-Based Multiprocessing Systems	Presentation	12
	<u>Books and References :</u> 1. Yu-Cheng Liu, Glenn A.Gibson, “Microcomputer Systems: The 8086 / 8088 Family – Architecture, Programming and Design”, Second Edition, Prentice Hall of India, 2007. 2. 8086/8088 family: Design Programming and Interfacing: John Uffenbeck ,PHI. 3. Hall, D.V, Microprocessor and Interfacing, Tata McGraw Hill Publishing Company, 2006		
	The following Programs/experiments are to be written for assembler and execute the same using 86 emulator 1. Programs for 16 bit arithmetic operations of 8086 (using various addressing modes) 2. Program for sorting an array for 8086 3. Program for searching for a number or character in a string for 8086 4. Program for string manipulations for 8086 5. Program for digital clock design using 8086 6. Interfacing ADC and DAC to 8086 7. Parallel communication between two microprocessors using 8255 8. Serial communication between two microprocessor kits using 8251 9. Interfacing to 8086 and programming got control stepper motor 10. Programming using arithmetic, logical and bit manipulation instructions of 8051 11. Program and verify Timer/ counter in 8051 12. Program and verify interrupt handling in 8051 13. UART operation in 8051 14. Communication between 8051 kit and PC 15. Interfacing LCD to 8051 16. Interfacing Matrix / Key board to 8051 Data Transfer from peripheral to memory through DMA controller 8237 / 8257		

Programme : B. Sc IT (Information Technology)					Semester : I		
Course : Notion of Operating System					Code : NMUBSCIT203		
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	25 Marks	75 Marks	20 Marks	30 Ma
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							
<ol style="list-style-type: none"> To learn Operating systems in depth, memory management , file system and system To learn Linux operating system, from installation to basic administration 							calls structures
Learning Outcomes							
<ol style="list-style-type: none"> Students will be able to understand the basics of operating system and its structure. Understanding memory management in OS Understanding File Systems in OS Commands of linux operating system and can write shell scripts Linux administrators role 							
Pedagogy							
Powerpoint Presentation , Computational Learning , Algorithm Solving.							

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	Operating system structure operating system concepts, system calls, operating system structure , Operating System advanced concept : Processes and Threads: Processes, threads, interprocess communication, scheduling, IPC problems. Deadlocks: Resources, introduction to deadlocks, the ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention issues.	Powerpoint presentation	12
II	Memory Management: No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, segmentation	Computational learning	12
III	File Systems: Files, directories, file system implementation, file-system management and optimization, MS-DOS file system, UNIX V7 file system, CD ROM file system	Powerpoint presentation, Algorithm Solving	12
IV	Introduction to Red Hat Enterprise Linux: Linux, Open Source and Red Hat, Origins of Linux, Distributions, Duties of Linux System Administrator. Introduction to Basic Linux Commands Command Line: Working with the Bash Shell, Getting the Best of Bash, Useful Bash Key Sequences, Working with Bash History, Performing Basic File System Management Tasks, Working with Directories, Piping and Redirection, Finding Files.	Powerpoint presentation	12
V	System Administration Tasks: Performing Job Management Tasks, System and Process Monitoring and Management, Managing Processes with ps, Sending Signals to Processes with the kill Command, Using top to Show Current System Activity, Managing Process Niceness, Scheduling Jobs, Mounting Devices, Working with Links, Creating Backups, Managing Printers Working with Users, Groups, and Permissions: Managing Users and Groups, Commands for User Management, Managing Passwords, Modifying and Deleting User Accounts	Powerpoint presentation	12
	Reference Books: 1. Modern Operating System by Andrew S. Tanenbaum 2. Maurice J. Bach, "Design of UNIX Operating System", PHI 3. Red Hat Enterprise Linux 6 Administration Sander van Vugt John Wiley and Sons 2013 4. Red hat Linux Networking and System Administration Terry Collings and Kurt Wall Wiley 3 rd		

	5. . Linux Administration: A Beginner's Guide Wale Soyinka TMH Fifth Edition		
	<p><u>List of Practical:-</u></p> <p>1 Windows operating system commands</p> <p>2 Linux commands: Working with Directories:</p> <p>a. pwd, cd, absolute and relative paths, ls, mkdir, rmdir, b. file, touch, rm, cp. mv, rename, head, tail, cat, tac, more, less, strings, chmod</p> <p>3.. Linux commands: Working with files:</p> <p>a. ps, top, kill, pkill, bg, fg, b. grep, locate, find, locate. c. date, cal, uptime, w, whoami, finger, uname, man, df, du, free, whereis, which. d. Compression: tar, gzip.</p> <p>4 Working with Linux Desktop and utilities</p> <p>a. The vi editor. b. Graphics c. Terminal d. Adjusting display resolution e. Using the browsers f. Configuring simple networking g. Creating users and shares</p>		

Programme : FY B. Sc IT (Information Technology)				Semester - II			
Course : Statistical Methods for Data Science				Code : NMUBSCIT204			
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+1	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							
<ul style="list-style-type: none"> • To learn mathematics basis for various data sciences • To learn big data concepts 							
Learning Outcomes							
<ul style="list-style-type: none"> • Ability to develop Mathematical Modeling for problem solving • Understanding Regression Model • Understanding Distribution Model • Understanding concept of Data Science • Awareness of Machine learning algorithms and techniques 							
Pedagogy							
PPT + Blackboard.							

Detailed Syllabus Plan			
Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	Mathematical Modeling and Engineering Problem Solving: A Simple Mathematical Model, Conservation Laws and Engineering Problems Numerical solution of 1st and 2nd order differential equations: Taylor series, Euler's Method, Modified Euler's Method, Runge-Kutta Method for 1st and 2 and Order Differential Equations	PPT +Blackboard	12
II	Correlation Analysis, Least-Squares Regression: Linear Regression, Polynomial Regression, Multiple Linear Regression, General Linear Least Squares, Nonlinear Regression Linear Programming: Linear optimization problem, Formulation and Graphical solution, Basic solution and Feasible solution. Random variables: Discrete and Continuous random variables, Probability density function, Probability distribution of random variables, Expected value, Variance.	PPT +Blackboard	12
III	Distributions: Discrete distributions: Uniform, Binomial, Poisson, Bernoulli, Continuous distributions: uniform distributions, exponential, (derivation of mean and variance only and state other properties and discuss their applications) Normal distribution state all the properties and its applications.	PPT +Blackboard	12
IV	Introduction: What is Data Science? - Big Data and Data Science hype – and getting past the hype - Why now? – Datafication - Current landscape of perspectives - Skill sets needed. Statistical Inference - Populations and samples - Statistical modeling, probability distributions, fitting a model - Intro to R Exploratory Data Analysis and the Data Science Process - Basic tools (plots, graphs and summary statistics) of EDA - Philosophy of EDA - The Data Science Process. Three Basic Machine Learning Algorithms - Linear Regression - k-Nearest Neighbors (k-NN) - k-means	PPT +Blackboard	12
V	Study of common Applications -: Filtering Spam Why Linear Regression and k-NN are poor choices for Filtering Spam - Naive Bayes and why it works for Filtering Spam - Data Wrangling: APIs and other	PPT +Blackboard	12

	<p>tools for scrapping the Web . Feature Generation and Feature Selection (Extracting Meaning From Data) - Application: user (customer) retention - Feature Generation (brainstorming, role of domain expertise, and place for imagination) - Feature Selection algorithms – Filters; Wrappers; Decision Trees; Random Forests</p>		
	<p>Reference Book 1.Introductory Methods of Numerical Methods S. S. Shastri 2. Data Science and Analytics by V.K. Jain 3. Statistics for Data Science by James D. Miller</p>		
	<p>List of Practical 1. Iterative Calculation a. Program for iterative calculation. b. Program to calculate the roots of a quadratic equation using the formula. c. Program to evaluate e^x using infinite series. 2. Solution of algebraic and transcendental equations: a. Program to solve algebraic and transcendental equation by bisection method. b. Program to solve algebraic and transcendental equation by false position method. c. Program to solve algebraic and transcendental equation by Secant method. d. Program to solve algebraic and transcendental equation by Newton Raphson method. 3. Interpolation a. Program for Newton’s forward interpolation. b. Program for Newton’s backward interpolation. c. Program for Lagrange’s interpolation. 4 Solving linear system of equations by iterative methods a. Program for solving linear system of equations using Gauss Jordan method. b. Program for solving linear system of equations using Gauss Seidel method. 5. Numerical Differentiation a. Programing to obtain derivatives numerically. 6. Numerical Integration a. Program for numerical integration using Trapezoidal rule. b. Program for numerical integration using Simpson’s 1/3rd rule. c. Program for numerical integration using Simpson’s 3/8th rule. 7. Solution of differential equations</p>		

	<ul style="list-style-type: none"> a. Program to solve differential equation using Euler's method b. Program to solve differential equation using modified Euler's method. c. Program to solve differential equation using Runge-kutta 2nd order and 4th order methods. <p>8. Regression</p> <ul style="list-style-type: none"> a. Program for Linear regression. b. Program for Polynomial Regression. c. Program for multiple linear regression. d. Program for non-linear regression. <p>9. Random variables and distributions</p> <ul style="list-style-type: none"> a. Program to generate random variables. b. Program to fit binomial distribution. c. Program to fit Poisson distribution. <p>10. Distributions</p> <ul style="list-style-type: none"> a. Program for Uniform distribution. b. Program for Bernoulli distribution c. Program for Negative binomial distribution 		
--	---	--	--

Programme : B. Sc IT (Information Technology)							Semester II	
Course : Digital Marketing							Code : NMUBSCIT205	
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	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								
<ul style="list-style-type: none"> To understand significance of Digital Marketing and its applications in Business and Various Sectors To provide an insight on Digital Marketing activities on various Social Media platforms and its emerging significance in Business To understand Latest Trends and Practices in E-Commerce and Digital Marketing, along with its Challenges and Opportunities for an Organization 								
Learning Outcomes								
<ol style="list-style-type: none"> Understanding about e-commerce, m-commerce, e business and digital marketing Understand Search Engine Optimization Email and Mobile marketing Understanding various software tools of DM Birds Eye view and future view of DM 								
Pedagogy								
<ol style="list-style-type: none"> Lecture Presentations. Assignments Group discussions Mini Project Industrial Visit Guest Lectures 								

Module	Module Content	Module wise Pedagogy Used	Duration of Module
I	Introduction to marketing concepts: definition & objectives of marketing , understanding marketing activities , P's of marketing & marketing mix Introduction to digital Marketing . Technology Behind DM. Concept of E-Commerce, M-Commerce and E-Business. Electronic Data Interchange (EDI) .Digital Marketing Strategy : Need , Defining. 4 Ps of Marketing and 10 Ps of Digital Marketing. Role of Web development in Digital Marketing.	Presentation , guest lecture, assignments	09
II	Search : Being found online. Search Engine Basics, Optimizing your website for SEO. Advertising on search engines, Mobile Search Understanding Social Media: Different forms of social media, Social media dashboards.	Presentation , guest lecture, assignments	09
III	Understanding e-mail marketing. Planning, measurement of success. Understanding mobile marketing :Market size, Mobile gaming and applications. Mobile privacy, Mobile data. Building Multichannel Marketing Strategy.	Presentation , guest lecture, assignments	09
IV	Various Software tools for Digital Marketing:- Marketo, Vocus, HubSpot, Yesware, Sailthru, Optimove, LocalVox, MailChimp. Mind Mapping in Digital Marketing. Digital Marketing Process.	Presentation , guest lecture, assignments	09
V	Future of Digital Marketing. Commercial Advantage, Censorship and privacy issues . Power of voice and thought Life without Google.	Presentation , guest lecture, assignments	09
	References/Books:		
	Term Project / Practical / Case Study 1. Listing down marketing activities in manufacturing and service organisations 2. Introduction to WIX Platform 3. Create a simple homepage using WIX. 4. Create a simple website using WIX 5. Create a FREE ecommerce Website using WIX. 6. Increase SEO Results as a Small Business 7. Develop Key Performance Indicators(KPI)		