Vidya Prasarak Mandal's
B. N. Bandodkar College of
Science (Autonomous), Thane
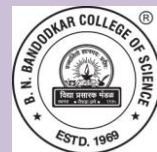
Syllabus for

Programme : Bachelor of Science

Specific Programme : Information Technology

[F.Y.B.Sc. (I.T.)]

Revised under Autonomy

From academic year 2021 - 2022

This page is intentionally left blank

Preamble

The B.Sc. Information Technology programme is aimed to make the students employable and impart industry oriented training. The main objectives of the course are:

➢ to think analytically, creatively and critically in developing robust, extensible and highly maintainable technological solutions to simple and complex problems.

➢ to apply their knowledge and skills to be employed and excel in IT professional careers and/or to continue their education in IT and/or related post graduate programmes.

➢ to be capable of managing complex IT projects with consideration of the human, financial and environmental factors.

➢ to work effectively as a part of a team to achieve a common stated goal.

➢ to adhere to the highest standards of ethics, including relevant industry and organizational codes of conduct.

➢ to communicate effectively with a range of audiences both technical and non-technical.

➢ to develop an aptitude to engage in continuing professional development.

The syllabus is aimed to achieve the objectives. The syllabus spanning three years covers the industry relevant courses. The students will be ready for the jobs available in different fields like:

➢ Software Development (Programming)

➢ Website Development

➢ Mobile app development

➢ Embedded Systems Programming

➢ Embedded Systems Development

➢ Software Testing

➢ Networking

➢ Database Administration

➢ System Administration

➢ Cyber Law Consultant

➢ GIS (Geographic Information Systems)

➢ IT Service Desk

➢ Security

And many others

The students will also be trained in communication skills and green computing.

Eligibility :

Passed 12th standard (HSC) of Maharashtra State Board / CBSE / ICSE board or equivalent with Mathematics as one of the subject.

Duration : 3 years

Mode of Conduct :

Laboratory practicals / Offline lectures / Online lectures

Program Specific Outcome

Identify, design, and analyze complex computer systems and implement and interpret the results from those systems. ... Select and apply current techniques, skills, and tools necessary for computing practice and integrate IT-based solutions into the user environment effectively.

VPM's B.N.Bandodkar College of Science (Autonomous), Thane

F.Y.B.Sc. (I.T.)

Structure of Programme

CourseCode	Course Title	No. of lectures	Credits
BNBUSIT1T1	Imperative Programming	45	2
BNBUSIT1T2	Digital Electronics	45	2
BNBUSIT1T3	Operating Systems	45	2
BNBUSIT1T4	Discrete Mathematics	45	2
BNBUSIT1T5	Communication Skills	45	2
BNBUSIT1P1	Imperative Programming Practical	30	2
BNBUSIT1P2	Digital Electronics Practical	30	2
BNBUSIT1P3	Operating Systems Practical	30	2
BNBUSIT1P4	Discrete Mathematics Practical	30	2
BNBUSIT1P5	Communication Skills Practical	30	2
Total		375	20

CourseCode	Course Title	No. of lectures	Credits
BNBUSIT2T1	Object oriented Programming	45	2
BNBUSIT2T2	Microprocessor Architecture	45	2
BNBUSIT2T3	Web Programming	45	2
BNBUSIT2T4	Numerical and Statistical Methods	45	2
BNBUSIT2T5	Green Computing	45	2
BNBUSIT2P1	Object Oriented Programming Practical	30	2
BNBUSIT2P2	Microprocessor Architecture Practical	30	2
BNBUSIT2P3	Web Programming Practical	30	2
BNBUSIT2P4	Numerical and Statistical Methods Practical	30	2
BNBUSIT2P5	Green Computing Practical	30	2
Total		375	20

Semester I

Course Code	Course Title	Credits	No. of lectures
BNBUSIT1T1	Imperative Programming	2	

Course Outcomes: Upon completion of this course, students will acquire knowledge about and able to
- Implement the algorithms
- Draw flowcharts for solving Mathematical and Engineering problems.
- Demonstrate an understanding of computer programming language concepts.
- To be able to develop C programs.

Unit I :	**Introduction: Need of writing programs,** Types of Programming languages, History, features and application. Simple program logic, program development cycle, pseudo code statements and flowchart symbols, sentinel value to end a program, programming and user environments, evolution of programming models., desirable program characteristics. **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. **Operators and Expressions:** Arithmetic operators, unary operators, relational and logical operators, assignment operators, assignment operators, the conditional operator, library functions.	15
Unit II :	**Data Input and output:** Single character input and output, entering input data, scanf function, printf function, gets and puts functions, interactive programming. **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 **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 functions, prototype of a function: foo1lal parameter list, return type, function call, block structure, passing arguments to a function: call by reference, call by value. **Program structure:** Storage classes, automatic variables, external variables, static variables, multifile programs, more library functions,	15
Unit III :	**Preprocessor:** Features, #define and #include, Directives and Macros **Arrays:** Definition, processing, passing arrays to functions, multidimensional arrays, arrays and strings. **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 **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.	15

Course Code	Course Title	Credits	No. of
BNBUSIT1T2	**Digital Electronics**	**2**	lectures

Learning Outcomes: Students would gain enough knowledge
1. Have a thorough understanding of the fundamental concepts and techniques used in digital electronics.
2. To understand and examine the structure of various number systems and its application in digital design.
3. The ability to understand, analyze and design various combinational circuits.
4. The ability to understand, analyze and design various sequential circuits

Unit I :	**History : Electronics, Devices, Facts and History, History of Digitization** **Number System:** Analog System, digital system, numbering system, binary number system, octal number system, hexadecimal number system, conversion from one number system to another, floating point numbers, weighted codes binary coded decimal, non-weighted codes Excess – 3 code, Gray code, Alphanumeric codes – ASCII Code, EBCDIC, ISCII Code, Hollerith Code, Morse Code, Teletypewriter (TTY), Error detection and correction, Universal Product Code, Code conversion. **Binary Arithmetic:** Binary addition, Binary subtraction, Negative number representation, Subtraction using 1's complement and 2's complement, Binary multiplication and division, Arithmetic in octal number system, Arithmetic in hexadecimal number system, BCD and Excess – 3 arithmetic. **Boolean Algebra and Logic Gates:** Introduction, Logic (AND OR NOT), Boolean theorems, Boolean Laws, De Morgan's Theorem, Perfect Induction, Reduction of Logic expression using Boolean Algebra, Deriving Boolean expression from given circuit, exclusive OR and Exclusive NOR gates, Universal Logic gates, Implementation of other gates using universal gates, Input bubbled logic, Assertion level.	**15**
Unit II :	**Minterm, Maxterm and Karnaugh Maps:** Introduction, minterms and sum of minterm form, maxterm and Product of maxterm form, Reduction technique using Karnaugh maps – 2/3/4/5/6 variable K-maps, Grouping of variables in K-maps, K-maps for product of sum form, minimize Boolean expression using K-map and obtain K-map from Boolean expression, Quine Mc Cluskey Method. **Combinational Logic Circuits:** Introduction, Multi-input, multi-output Combinational circuits, Code converters design and implementations **Arithmetic Circuits:** Introduction, Adder, BCD Adder, Excess – 3 Adder, Binary Subtractors, BCD Subtractor, Multiplier, Comparator. **Multiplexer, Demultiplexer, ALU, Encoder and Decoder:** Introduction, Multiplexer, Demultiplexer, Decoder, ALU, Encoders.	**15**
Unit III :	**Sequential Circuits: Flip-Flop:** Introduction, Terminologies used, S-R flip-flop, D flip-fop, JK flip- flop, Race-around condition, Master – slave JK flip-flop, T flip-flop, conversion from one type of flip-flop to another, Application of flip- flops. **Counters:**Introduction, Asynchronous counter, Terms related to counters, IC7493 (4-bit binary counter), Synchronous counter, Bushing, Type T Design, Type JK Design, Pre-settable counter, IC7490, IC7492, Synchronous counter ICs, Analysis of counter circuits. **Shift Register:** Introduction, parallel and shift registers, serial shifting, serial–in serial– out, serial–in parallel–out , parallel–in parallel–out, Ring counter, Johnson counter, Applications of shift registers, Pseudo-random binary sequence generator, IC7495, Seven Segment displays, analysis of shift counters.	**15**

Course Code BNBUSIT1T3	Course Title Operating Systems	Credits 2	No. of lectures

Learning Outcomes: A successful student will be able to understand the
- basic components of a computer operating system
- the interactions among the various components,
- policies for scheduling, deadlocks, memory management
- Synchronization, system calls, and files systems.

Unit I :	**Timeline of Operating Systems, How to program your very own Operating system** **Introduction:** What is an operating system? History of operating system, computer hardware, different operating systems, operating system concepts, system calls, operating system structure. **Processes and Threads**: Processes, threads, inter process communication, scheduling, IPC problems. **Memory Management:** No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, segmentation.	15
Unit II :	**File Systems:** Files, directories, file system implementation, file-system management and optimization, MS-DOS file system, UNIX V7 file system, CD ROM file system. **Input-Output:** Principles of I/O hardware, Principles of I/O software, I/O software layers, disks, clocks, user interfaces: keyboard, mouse, monitor, thin clients, power management, **Deadlocks:** Resources, introduction to deadlocks, the ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention, issues. **Multiple Processor Systems** Multiprocessors, multicomputers, distributed systems.	15
Unit III :	**Virtualization and Cloud:** History, requirements for virtualization, type 1 and 2 hypervisors, techniques for efficient virtualization, hypervisor microkernels, memory virtualization, I/O virtualization, Virtual appliances, virtual machines on multicore CPUs, Clouds. **Case Study on LINUX and ANDROID:** History of Unix and Linux, Linux Overview, Processes in Linux, Memory management in Linux, I/O in Linux, Linux file system, security in Linux. Android **Case Study on Windows:** History of windows through Windows 10, programming windows, system structure, processes and threads in windows, memory management, caching in windows, I/O in windows, Windows NT file system, Windows power management, Security in windows.	15

Course Code	Course Title	Credits	No. of lectures
BNBUSIT1T4	**Discrete Mathematics**	**2**	

Learning Outcomes : After completing this course satisfactorily, a student will:
- Be able to construct simple mathematical proofs and possess the ability to verify them
- Have substantial experience to comprehend formal logical arguments
- Be skillful in expressing mathematical properties formally via the formal language of propositional logic and predicate logic

Unit I :	**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, Russell's Paradox and the Halting Problem. **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	**15**
Unit II :	**Elementary Number Theory and Methods of Proof**: Introduction to Direct Proofs, 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. **Sequences, Mathematical Induction, and Recursion**: Sequences, Mathematical Induction, Strong Mathematical Induction and the Well- Ordering Principle for the Integers, Correctness of algorithms, 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	**15**
Unit III :	**Relations**: Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations, Partial Order Relations **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. **Counting and Probability:** Introduction, Possibility Trees and the Multiplication Rule, Possibility Trees and the Multiplication Rule, Counting Elements of Disjoint Sets: The Addition Rule, The Pigeonhole Principle, Counting Subsets of a Set: Combinations, r- Combinations with Repetition Allowed, Probability Axioms and Expected Value, Conditional Probability, Bayes' Formula, and Independent Events.	**15**

Course Code	Course Title	Credits	No. of lectures
BNBUSIT1T5	**Communication Skills**	**2**	

On completion of the course, student will be able to–
- Effectively communicate through verbal/oral communication and improve the listening skills \
- Write precise briefs or reports and technical documents .
- Actively participate in group discussion / meetings / interviews and prepare & deliver presentations .
- Become more effective individual through goal/target setting, self motivation and creative thinking.

Unit I :	**Introduction to communication skills, Need for Communication, Process of Communication** **The Seven Cs of Effective Communication:** Completeness, Conciseness, Consideration, Concreteness, Clarity, Courtesy, Correctness **Understanding Business Communication:** Nature and Scope of Communication, Non-verbal Communication, Cross-cultural communication, Technology-enabled Business Communication **Writing Business Messages and Documents:** Business writing, Business Correspondence, Instructions Business Reports and Proposals	**15**
Unit II :	**Developing Oral Communication Skills for Business:** Effective Listening, Business Presentations and Public Speaking, Conversations, Interviews **Developing Oral Communication Skills for Business:** Meetings and Conferences, Group Discussions and Team Presentations, TeamBriefing, **Understanding Specific Communication Needs:** Communication across Functional Areas	**15**
Unit III :	**Understanding Specific Communication Needs:** Corporate Communication, Persuasive Strategies in Business Communication, Ethics in Business Communication, Business Communication Aids **Report writing : How to write Project reports, importance of project report, Tables of Contents, Executive Summaries, Feasibility Reports, Investigative Reports, Laboratory Reports, Test Reports** **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.	**15**

Course Code BNBUSIT1P1	Course Title Imperative Programming Practical	Credits 2	No. of lectures
Practical 1	**Basic Programs:**		**3**
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.		
Practical 2	**Programs on variables:**		**3**
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.		
Practical 3	**Conditional statements and loops(basic)**		**3**
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.		
c.	Write a program to check whether the number is positive, negative or zero.		
d.	Write a program to find the factorial of a number.		
e.	Write a program to check whether the entered number is prime or not.		
f.	Write a program to find the largest of three numbers.		
Practical 4	**Conditional statements and loops(advanced)**		**3**
a.	Write a program to find the sum of squares of digits of a number.		
b.	Write a program to reverse the digits of an integer.		
c.	Write a program to find the sum of numbers from 1 to 100.		
d.	Write a programs to print the Fibonacci series.		
e.	Write a program to find the reverse of a number.		
f.	Write a program to find whether a given number is palindrome or not.		
g.	Write a program that solve the quadratic equation $$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$		
h.	Write a program to check whether the entered number is Armstrong or not.		
i.	Write a program to count the digit in a number		

Practical 5	Programs on patterns:	3
a.	Programs on different patterns.	
Practical 6	Functions:	3
a.	Programs on Functions.	
Practical 7	Recursive functions	3
a.	Write a program to find the factorial of a number using recursive function.	
b.	Write a program to find the sum of natural number using recursive function.	
Practical 8	Arrays	3
a.	Write a program to find the largest value that is stored in the array.	
b.	Write a program using pointers to compute the sum of all elements stored in an array.	
c.	Write a program to arrange the 'n' numbers stored in the array in ascending and descending order.	
d.	Write a program that performs addition and subtraction of matrices.	
e.	Write a program that performs multiplication of matrices.	
Practical 9	Pointers	3
a.	Write a program to demonstrate the use of pointers.	
b.	Write a program to perform addition and subtraction of two pointer variables.	
Practical 10	Structures and Unions	3
a.	Programs on structures.	
b.	Programs on unions.	

Course Code BNBUSIT1P2	Course Title Digital Electronics Practical	Credits 2	No. of lectures
Practical 1	Study of Logic gates and their ICs and universal gates:		3
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.		
Practical 2	Implement the given Boolean expressions using minimum number of gates.		3
a.	Verifying De Morgan's laws.		
b.	Implement other given expressions using minimum number of gates.		
c.	Implement other given expressions using minimum number of ICs.		

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

Practical 9		3
a.	Study of IC 7490, 7492, 7493 and designing mod-n counters using these.	
b.	Designing mod-n counters using IC 7473 and 7400 (NAND gates)	
Practical 10		3
a.	Design serial – in serial – out, serial – in parallel – out, parallel – in serial – out, parallel – in parallel – out and bidirectional shift registers using IC 7474.	
b.	Study of ID 7495.	
c.	Implementation of digits using seven segment displays.	

Course Code BNBUSIT1P3	Course Title **Operating Systems Practical**	Credits **2**	No. of lectures
Practical 1	Installation of virtual machine software.		**3**
Practical 2	Installation of Linux operating system (RedHat / Ubuntu) on virtual machine.		**3**
Practical 3	Installation of Windows operating system on virtial machine.		**3**
Practical 4	**Linux commands: Working with Directories:**		**3**
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		
Practical 5	**Linux commands: Working with files:**		**3**
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.		
Practical 6	**Windows (DOS) Commands – 1**		**3**
a.	Date, time, prompt, md, cd, rd, path.		
b.	Chkdsk, copy, xcopy, format, fidsk, cls, defrag, del, move.		
Practical 7	**Windows (DOS) Commands – 2**		**3**
a.	Diskcomp, diskcopy, diskpart, doskey, echo		
b.	Edit, fc, find, rename, set, type, ver		
Practical 8	**Working with Windows Desktop and utilities**		**3**
a.	Notepad		
b.	Wordpad		

c.	Paint	
d.	Taskbar	
e.	Adjusting display resolution	
f.	Using the browsers	
g.	Configuring simple networking	
h.	Creating users and shares	
Practical 9	**Working with Linux Desktop and utilities**	**3**
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	
Practical 10	**Installing utility software on Linux and Windows**	**3**

Course Code BNBUSIT1P4	Course Title Discrete Mathematics Practical	Credits 2	No. of lectures
Practical 1	**Set Theory**		**3**
a.	Inclusion Exclusion principle.		
b.	Power Sets		
c.	Mathematical Induction		
Practical 2	**Functions and Algorithms**		**3**
a.	Recursively defined functions		
b.	Cardinality		
c.	Polynomial evaluation		
d.	Greatest Common Divisor		
Practical 3	**Counting**		**3**
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 partitions	
j.	Unordered partitions	
Practical 4	**Probability Theory**	**3**
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	
Practical 5	**Graph Theory**	**3**
a.	Paths and connectivity	
b.	Minimum spanning tree	
c.	Isomorphism	
Practical 6	**Directed Graphs**	**3**
a.	Adjacency matrix	
b.	Path matrix	
Practical 7	**Properties of integers**	**3**
a.	Division algorithm	
b.	Primes	
c.	Euclidean algorithm	
d.	Fundamental theorem of arithmetic	
e.	Congruence relation	
f.	Linear congruence equation	
Practical 8	**Algebraic Systems**	**3**

a.	Properties of operations		
b.	Roots of polynomials		
Practical 9	**Boolean Algebra**		**3**
a.	Basic definitions in Boolean Algebra		
b.	Boolean algebra as lattices		
Practical 10	**Recurrence relations**		**3**
a.	Linear homogeneous recurrence relations with constant coefficients		
b.	Solving linear homogeneous recurrence relations with constant coefficients		
c.	Solving general homogeneous linear recurrence relations		

Course Code BNBUSIT1P5	Course Title **Communication Skills Practical**	Credits 2	No. of lectures
Practical 1	Communication Origami, Guessing Game, Guessing the emotion		**3**
Practical 2	Body Language, Follow All Instructions, Effective Feedback Skills		**3**
Practical 3	The Name Game, Square Talk (Effective Communication), Room 101 (Influential and persuasive skills)		**3**
Practical 4	Back to Back Communication, Paper Shapes (Importance of two-way communication), Memory Test(Presentation Skills)		**3**
Practical 5	Exercises on Communication Principles		**3**
Practical 6	Exercises on communication icebreakers		**3**
Practical 7	Communication exercises		**3**
	For the following practicals, Microsoft Office, Open Office, Libre Office or any other software suite can be used.		
Practical 8	Use of word processing tools for communication		**3**
Practical 9	Use of spreadsheet tools for communication		**3**
Practical 10	Use of presentation tools for communication		**3**

References

Course Code	Course Title
BNBUSIT1T1	**Imperative Programming**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Programming with C	Byron Gottfried	Tata McGRAW-Hill	2nd	1996
2.	Programming Logic and Design	Joyce Farell	Cengage Learning	8th	2014
3.	"C" Programming"	Brian W. Kernighan and Denis M. Ritchie.	PHI	2nd	
4.	Let us C	Yashwant P. Kanetkar,	BPB publication		
5.	C for beginners	Madhusudan, Mothe	X-Team Series	1st	2008
6.	21st Century C	Ben Klemens	OReilly	1st	2012

Course Code	Course Title
BNBUSIT1T2	**Digital Electronics**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Digital Electronics and Logic Design	N. G. Palan	Technova		
2.	Make Electronics	Charles Platt	O'Reilly	1st	2010
3.	Modern Digital Electronics	R. P. Jain	Tata McGraw Hill	3rd	
4.	Digital Principles and Applications	Malvino and Leach	Tata McGraw Hill		
5.	Digital Electronics: Principles, Devices and Applications,	Anil K. Maini	Wiley		2007

Course Code	Course Title
BNBUSIT1T3	**Operating Systems**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Modern Operating Systems	Andrew S. Tanenbaum, Herbert Bos	Pearson	4th	2014
2.	Operating Systems – Internals and Design Principles	Willaim Stallings	Pearson	8th	2009
3.	Operating System Concepts	Abraham Silberschatz, Peter B. Galvineg Gagne	Wiley	8th	
4.	Operating Systems	Godbole and Kahate	McGraw Hill	3rd	

Course Code	Course Title
BNBUSIT1T4	**Discrete Mathematics**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Discrete Mathematics with Applications	Sussana S. Epp	Cengage Learning	4th	2010
2.	Discrete Mathematics, Schaum's Outlines Series	Seymour Lipschutz, Marc Lipson	Tata MCGraw Hill		2007
3.	Discrete Mathematics and its Applications	Kenneth H. Rosen	Tata MCGraw Hill		
4.	Discrete mathematical structures	B Kolman RC Busby, S Ross	PHI		
5.	Discrete structures	Liu	Tata MCGraw Hill		

Course Code	Course Title
BNBUSIT1T5	**Communication Skills**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Business Communication	Edited by Meenakshi Raman and Prakash Singh	Oxford University Press	Second	

Semester II

Course Code	Course Title	Credits	No. of lectures
BNBUSIT2T1	**Object Oriented Programming**	**2**	

On completion of the course, student will be able to–
• Articulate the principles of object-oriented problem solving and programming.
• Explain programming fundamentals, including statement and control flow and recursion.
• Apply the concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism.
• Program with basic data structures using array
• Program using objects and data abstraction, class, and methods in function abstraction.

Unit I :	**History of Programming, programming approaches, how to make your program readable and understandable** **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. **Principles of OOPS:** OOPS Paradigm, Basic Concepts of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing **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.	**15**
Unit II :	**Constructors and Destructors:** Introduction, Default Constructor, Parameterized Constructor and examples, Destructors **Polymorphism:** Concept of function overloading, overloaded operators, overloading unary and binary operators, overloading comparison operator, overloading arithmetic assignment operator, Data Conversion between objects and basic types, **Virtual Functions:** Introduction and need, Pure Virtual Functions, Static Functions, this Pointer, abstract classes, virtual destructors. **Exception Handling:** Introduction, Exception Handling Mechanism, Concept of throw & catch with example	**15**
Unit III :	**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. **Templates:** Introduction, Function Template and examples, Class Template and examples. **Working with Files:** Introduction, File Operations, Various File Modes, File Pointer and their Manipulation	**15**

Course Code BNBUSIT2T2	Course Title Microprocessor Architecture	Credits 2	No. of lectures

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

- Assess and solve basic binary math operations using the microprocessor
- explain the microprocessor's internal architecture and its operation.
- addressing modes and data transfer instructions of the target microprocessor and microcontroller.
- Analyze assembly language programs; select appropriate assemble into machine a cross assembler utility of a microprocessor

Unit I :	**Evolution of Microprocessor, Block diagram of Microprocessor** **Microprocessor, microcomputers and Assembly Language:** Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications. **Microprocessor Architecture and Microcomputer System:** Microprocessor Architecture and its operation's, Memory, I/O Devices, Microcomputer System, Logic Devices and Interfacing, Microprocessor-Based System Application. **8085 Microprocessor Architecture and Memory Interface:** Introduction, 8085 Microprocessor unit, 8085-Based Microcomputer, Memory Interfacing, Interfacing the 8155 Memory Segment, Illustrative Example: Designing Memory for the MCTS Project, Testing and Troubleshooting Memory Interfacing Circuit, 8085-Based Single-Board microcomputer. **Interfacing of I/O Devices :** Basic Interfacing concepts, Interfacing Output Displays, Interfacing Input Devices, Memory Mapped I/O, Testing and Troubleshooting I/O Interfacing Circuits. **Introduction to 8085 Assembly Language Programming:** The 8085 Programming Model, Instruction Classification, Instruction, Data and Storage, Writing assembling and Execution of a simple program, Overview of 8085 Instruction Set, Writing and Assembling Program.	**15**
Unit II :	**Introduction to 8085 Instructions:** Data Transfer Operations, Arithmetic Operations, Logic Operation, Branch Operation, Writing Assembly Languages Programs, Debugging a Program. **Programming Techniques With Additional Instructions:** Programming Techniques: Looping, Counting and Indexing, Additional Data Transfer and 16-Bit Arithmetic Instructions, rithmetic Instruction Related to Memory, Logic Operations: Rotate, Logics Operations: Compare, Dynamic ebugging. **Counters and Time Delays:** Counters and Time Delays, Illustrative Program: Hexadecimal Counter, Illustrative Program: zero-to-nine (Modulo Ten) Counter, Generating Pulse Waveforms, Debugging Counter and Time-Delay Programs. **Stacks and Sub-Routines:** Stack, Subroutine, Restart, Conditional Call, Return Instructions, Advanced Subroutine concepts. **Code Conversion, BCD Arithmetic, and 16-Bit Data Operations:** BCD-to-Binary Conversion, Binary-to-BCD Conversion, BCD-to-Seven-Segment-LED Code Conversion, Binary-to-ASCII and ASCII- to-Binary Code Conversion, BCD Addition, BCD Subtraction, Introduction, To advanced Instructions and Applications, Multiplication, Subtraction With Carry.	**15**
Unit III :	**Software Development System and Assemblers:** Microprocessors-Based Software Development system, Operating System and Programming Tools, Assemblers and Cross-Assemblers, Writing Program Using Cross Assemblers. **Interrupts:** The 8085 Interrupt, 8085 Vectored Interrupts, Restart as S/W Instructions, Additional I/O Concepts and processes. **The Pentium and Pentium Pro microprocessors:** Introduction, Special Pentium	**15**

	registers, Memory management, Pentium instructions, Pentium Pro microprocessor, Special Pentium Pro features. **Core 2 and later Microprocessors:** Introduction, Pentium II software changes, Pentium IV and Core 2, i3, i5 and i7. **SUN SPARC Microprocessor:** Architecture, Register file, data types and instruction format	

Course Code	Course Title	Credits	No. of lectures
BNBUSIT2T3	**Web Programming**	**2**	

Upon completion of this course, the students will be able to:
- Describe fundamentals of web
- Introduce the creation of webpage
- write well-structured, easily maintained, standards-compliant, accessible code
- Build Dynamic web site using server side Programming

Unit I :	**Internet and the World Wide Web:** What is Internet? Introduction to internet and its applications, E-mail, 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 **HTML5:** Introduction, Why HTML5? Formatting text by using tags, using lists and backgrounds, Creating hyperlinks and anchors. Style sheets, CSS formatting text using style sheets, formatting paragraphs using style sheets. **HTML5 Page layout and navigation:** Creating navigational aids: planning site organization, creating text based navigation bar, creating graphics based navigation bar, creating graphical navigation bar, creating image map, redirecting to another URL, creating division based layouts: HTML5 semantic tags, creating divisions, creating HTML5 semantic layout, positioning and formatting divisions. **HTML5 Tables, Forms and Media:** Creating tables: creating simple table, specifying the size of the table, specifying the width of the column, merging table cells, using tables for page layout, formatting tables: applying table borders, applying background and foreground fills, changing cell padding, spacing and alignment, creating user forms: creating basic form, using check boxes and option buttons, creating lists, additional input types in HTML5, Incorporating sound and video: audio and video in HTML5, HTML multimedia basics, embedding video clips, incorporating audio on web page.	**15**
Unit II :	**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	**15**
Unit III :	**Document and its associated objects:** document, Link, Area, Anchor, Image, Applet, Layer **Events and Event Handlers :** General Information about Events, Defining Event Handlers, event, onAbort, onBlur, onChange, onClick, onDblClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload **PHP:** Why PHP and MySQL? Server-side scripting, PHP syntax and variables, comments, types, control structures, branching, looping, termination, functions, passing information with PHP, GET, POST, formatting form variables, superglobal arrays, strings and string functions, regular expressions, arrays, number handling,	**15**

	basic PHP errors/problems **Advanced PHP & MySQL:** PHP/MySQL Functions, Integrating web forms and databases, Displaying queries in tables, Building forms from queries, String and Regular Expressions, Sessions, Cookies and HTTP, E-Mail	

Course Code	Course Title	Credits	No. of lectures
BNBUSIT2T4	**Numerical and Statistical Methods**	**2**	

Upon successful completion of this course, students will be able to:
• Organize, manage and present data.
• Analyze statistical data graphically using frequency distributions and cumulative frequency distributions.
• Analyze statistical data using measures of central tendency, dispersion and location.
• Use the basic probability rules, including additive and multiplicative laws, using the terms, independent and mutually exclusive events.

Unit I :	**Mathematical Modeling and Engineering Problem Solving:** A Simple Mathematical Model, Conservation Laws and Engineering Problems **Approximations and Round-Off Errors:** Significant Figures, Accuracy and Precision, Error Definitions, Round-Off Errors **Truncation Errors and the Taylor Series:** The Taylor Series, Error Propagation, Total Numerical Errors, Formulation Errors and Data Uncertainty **Solutions of Algebraic and Transcendental Equations:** The Bisection Method, The Newton-Raphson Method, The Regula-falsi method, The Secant Method.	**15**
Unit II :	**Interpolation:** Forward Difference, Backward Difference, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Lagrange's Interpolation. **Solution of simultaneous algebraic equations (linear) using iterative methods:** Gauss-Jordan Method, Gauss-Seidel Method. **Numerical differentiation and Integration:** Numberical differentiation, Numerical integration using Trapezoidal Rule, Simpson's $1/3^{rd}$ and $3/8^{th}$ rules. **Numerical solution of 1st and 2nd order differential equations:** Taylor series, Euler's Method, Modified Euler's Method, Runge-Kutta Method for 1st and 2nd Order Differential Equations.	**15**
Unit III :	**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. **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.	**15**

Course Code	Course Title	Credits	No. of lectures
BNBUSIT2T5	**Green Computing**	2	

Upon successful completion of this course, students will be able to:
- Describe awareness among stakeholders and promote green agenda and green initiatives in their working environments leading to green movement
- Identify IT Infrastructure Management and Green Data Centre Metrics
- Illustrate various green IT services and its roles
- Use Green IT Strategies and metrics for ICT development

Unit I :	**Importance of learning Green Computing** **Overview and Issues:** Problems: Toxins, Power Consumption, Equipment Disposal, Company's Carbon Footprint: Measuring, Details, reasons to bother, Plan for the Future, Cost Savings: Hardware, Power. **Initiatives and Standards:** Global Initiatives: United Nations, Basel Action Network, Basel Convention, North America: The United States, Canada, Australia, Europe, WEEE Directive, RoHS, National Adoption, Asia: Japan, China, Korea. **Minimizing Power Usage:** Power Problems, Monitoring Power Usage, Servers, Low-Cost Options, Reducing Power Use, Data De-Duplication, Virtualization, Management, Bigger Drives, Involving the Utility Company, Low- Power Computers, PCs, Linux, Components, Servers, Computer Settings, Storage, Monitors, Power Supplies, Wireless Devices, Software.	15
Unit II :	**Cooling:** Cooling Costs, Power Cost, Causes of Cost, Calculating Cooling Needs, Reducing Cooling Costs, Economizers, On-Demand Cooling, HP's Solution, Optimizing Airflow, Hot Aisle/Cold Aisle, Raised Floors, Cable Management, Vapour Seal, Prevent Recirculation of Equipment Exhaust, Supply Air Directly to Heat Sources, Fans, Humidity, Adding Cooling, Fluid Considerations, System Design, Datacentre Design, Centralized Control, Design for Your Needs, Put Everything Together. **Changing the Way of Work:** Old Behaviours, starting at the Top, Process Reengineering with Green in Mind, Analysing the Global Impact of Local Actions, Steps: Water, Recycling, Energy, Pollutants, Teleworkers and Outsourcing, Telecommuting, Outsourcing, how to Outsource. **Going Paperless:** Paper Problems, The Environment, Costs: Paper and Office, Practicality, Storage, Destruction, Going Paperless, Organizational Realities, Changing Over, Paperless Billing, Handheld Computers vs. the Clipboard, Unified Communications, Intranets, What to Include, Building an Intranet, Microsoft Office SharePoint Server 2007, Electronic Data Interchange (EDI), Nuts and Bolts, Value Added Networks, Advantages, Obstacles. **Recycling:** Problems, China, Africa, Materials, Means of Disposal, Recycling, Refurbishing, Make the Decision, Life Cycle, from beginning to end, Life, Cost, Green Design, Recycling Companies, Finding the Best One, Checklist, Certifications, Hard Drive Recycling, Consequences, cleaning a Hard Drive, Pros and cons of each method, CDs and DVDs, good and bad about CD and DVDs disposal, Change the mind-set, David vs. America Online	15
Unit III :	**Hardware Considerations:** Certification Programs, EPEAT, RoHS, Energy Star, Computers, Monitors, Printers, Scanners, All-in-Ones, Thin Clients, Servers, Blade Servers, Consolidation, Products, Hardware Considerations, Planned Obsolescence, Packaging, Toxins, Other Factors, Remote Desktop, Using Remote Desktop, Establishing a Connection, In Practice **Greening Your Information Systems:** Initial Improvement Calculations, Selecting Metrics, Tracking Progress, Change Business Processes, Customer Interaction, Paper Reduction, Green Supply Chain, Improve Technology Infrastructure, Reduce PCs and	15

| | Servers, Shared Services, Hardware Costs, Cooling.
Staying Green: Organizational Check-ups, Chief Green Officer, Evolution, Sell the CEO, SMART Goals, Equipment Check-ups, Gather Data, Tracking the data, Baseline Data, Benchmarking, Analyse Data, Conduct Audits, Certifications, Benefits, Realities, Helpful Organizations. | |

Course Code BNBUSIT2P1	Course Title Object Oriented Programming Practical	Credits 2	No. of lectures
Practical 1	**Classes and methods**		**3**
a.	Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used repectively. Where getInfo() will be private method		
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.		
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.		
d.	Write a program to demonstrate function definition outside class and ccessing class members in function definition.		
Practical 2	**Using friend functions.**		**3**
a.	Write a friend function for adding the two complex numbers, using a single class		
b.	Write a friend function for adding the two different distances and display its sum, using two classes.		
c.	Write a friend function for adding the two matrix from two different classes and display its sum.		
Practical 3	**Constructors and method overloading.**		**3**
a.	Design a class Complex for adding the two complex numbers and also show the use of constructor.		
b.	Design a class Geometry containing the methods area() and volume() and also overload the area() function .		
c.	Design a class StaticDemo to show the implementation of static variable and static function.		
Practical 4	**Operator Overloading**		**3**
a.	Overload the operator unary(-) for demonstrating operator overloading.		
b.	Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.		
c.	Overload the + for concatenating the two strings. For e.g "Py" + "thon" = python		
Practical 5	**Inheritance**		**3**
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.		
Practical 6	**Virtual functions and abstract classes**		**3**
a.	Implement the concept of method overriding.		
b.	Show the use of virtual function		

c.	Show the implementation of abstract class.	
Practical 7	**String handling**	**3**
a.	String operations for string length , string concatenation	
b.	String operations for string reverse, string comparison,	
c.	Console formatting functions.	
Practical 8	**Exception handling**	**3**
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.	
Practical 9	**File handling**	**3**
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	
Practical 10	**Templates**	**3**
a.	Show the implementation for the following	
b.	Show the implementation of template class library for swap function.	
c.	Design the template class library for sorting ascending to descending and vice-versa	

Course Code	Course Title	Credits	No. of
BNBUSIT2P2	**Microprocessor Architecture Practical**	**2**	**lectures**
Practical 1	**Perform the following Operations related to memory locations.**		**3**
a.	Store the data byte 32H into memory location 4000H.		
b.	Exchange the contents of memory locations 2000H and 4000H		
Practical 2	**Simple assembly language programs.**		**3**
a.	Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.		
b.	Subtract two 8-bit numbers.		
c.	Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.		
d.	Add the contents of memory locations 40001H and 4001H and place the result in the memory locations 4002Hand 4003H.		

e.	Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.	
f.	Find the l's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.	
g.	Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.	
Practical 3	**Packing and unpacking operations.**	**3**
a.	Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H.	
b.	Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit.	
Practical 4	**Register Operations.**	**3**
a.	Write a program to shift an eight bit data four bits right. Assume that data is in register C.	
b.	Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair	
c.	Write a set of instructions to alter the contents of flag register in 8085.	
d.	Write a program to count number of l's in the contents of D register and store the count in the B register.	
Practical 5	**Multiple memory locations.**	**3**
a.	Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H	
b.	Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H.	
c.	Divide 16 bit number stored in memory locations 2200H and 2201H by the 8 bit number stored at memory location 2202H. Store the quotient in memory locations 2300H and 2301H and remainder in memory locations 2302H and 2303H.	
d.	Find the number of negative elements (most significant bit 1) in a block of data. The length of the block is in memory location 2200H and the block itself begins in memory location 2201H. Store the number of negative elements in memory location 2300H	
e.	Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself starts from memory location 2201H. Store the maximum number in memory location 2300H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.	
Practical 6	**Calculations with respect to memory locations.**	**3**
a.	Write a program to sort given 10 numbers from memory location 2200H in	

	the ascending order.	
b.	Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2Sample problem:	
c.	Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H.	
d.	Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H	
e.	Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H	
f.	Two decimal numbers six digits each, are stored in BCD package form. Each number occupies a sequence of byte in the memory. The starting address of first number is 6000H Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H	
g.	Add 2 arrays having ten 8-bit numbers each and generate a third array of result. It is necessary to add the first element of array 1 with the first element of array-2 and so on. The starting addresses of array l, array2 and array3 are 2200H, 2300H and 2400H, respectively	
Practical 7	**Assembly programs on memory locations.**	**3**
a.	Write an assembly language program to separate even numbers from the given list of 50 numbers and store them in the another list starting from 2300H. Assume starting address of 50 number list is 2200H	
b.	Write assembly language program with proper comments for the following: A block of data consisting of 256 bytes is stored in memory starting at 3000H. This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.	
c.	Add even parity to a string of 7-bit ASCII characters. The length of the string is in memory location 2040H and the string itself begins in memory location 2041H. Place even parity in the most significant bit of each character.	
d.	A list of 50 numbers is stored in memory, starting at 6000H. Find number of negative, zero and positive numbers from this list and store these results in memory locations 7000H, 7001H, and 7002H respectively	
e.	Write an assembly language program to generate fibonacci number.	
f.	Program to calculate the factorial of a number between 0 to 8.	
Practical 8	**String operations in assembly programs.**	**3**
a.	Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters	
b.	Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters.	
c.	Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsigned number in memory location 2201H. Store the 8 least significant bits	

	of the result in memory location 2300H and the 8 most significant bits in memory location 2301H.	
d.	Divide the 16-bit unsigned number in memory locations 2200H and 2201H (most significant bits in 2201H) by the B-bit unsigned number in memory location 2300H store the quotient in memory location 2400H and remainder in 2401H	
e.	DAA instruction is not present. Write a sub routine which will perform the same task as DAA.	
Practical 9	**Calculations on memory locations.**	**3**
a.	To test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 40FFH to 40FFH. In case of any error, it is indicated by writing 01H at port 10	
b.	Arrange an array of 8 bit unsigned no in descending order	
c.	Transfer ten bytes of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300H	
d.	Write a program to find the Square Root of an 8 bit binary number. The binary number is stored in memory location 4200H and store the square root in 4201H.	
e.	Write a simple program to Split a HEX data into two nibbles and store it in memory	
Practical 10	**Operations on BCD numbers.**	**3**
a.	Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit.	
b.	Subtract the BCD number stored in E register from the number stored in the D register	
c.	Write an assembly language program to multiply 2 BCD numbers	

Course Code BNBUSIT2P3	Course Title Web Programming Practical	Credits 2	No. of lectures
Practical 1	**Use of Basic Tags**	**3**	
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		
Practical 2	**Image maps, Tables, Forms and Media**	**3**	
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.	
Practical 3	**Java Script**	**3**
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.	
c.	Write a JavaScript program to display all the prime numbers between 1 and 100.	
a.	Write a JavaScript program to accept a number from the user and display the sum of its digits.	
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).	
e.	Write a java script program to design simple calculator.	
Practical 4	**Control and looping statements and Java Script references**	**3**
a.	Design a web page demonstrating different conditional statements.	
b.	Design a web page demonstrating different looping statements.	
c.	Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp).	
Practical 5	**Basic PHP I**	**3**
a.	Write a PHP Program to accept a number from the user and print it factorial.	
b.	Write a PHP program to accept a number from the user and print whether it is prime or not.	
Practical 6	**Basic PHP II**	**3**
a.	Write a PHP code to find the greater of 2 numbers. Accept the no. from the user.	
b.	Write a PHP program to display the following Binary Pyramid: 1 0 1 1 0 1 0 1 0 1	
Practical 7	**String Functions and arrays**	**3**
a.	Write a PHP program to demonstrate different string functions.	
b.	Write a PHP program to create one dimensional array.	
Practical 8	**PHP and Database**	**3**
a.	Write a PHP code to create: • Create a database College • Create a table Department (Dname, Dno, Number_Of_faculty)	
b.	Write a PHP program to create a database named "College". Create a table named "Student" with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 to 75 in a tabular format.	
c.	Design a PHP page for authenticating a user.	

Practical 9	Email	3
a.	Write a program to send email with attachment.	
Practical 10	Sessions and Cookies	3
a.	Write a program to demonstrate use of sessions and cookies.	

Course Code	Course Title	Credits	No. of lectures
BNBUSIT2P4	**Numerical and Statistical Methods Practical**	**2**	
Practical 1	**Iterative Calculation**		3
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.		
Practical 2	**Solution of algebraic and transcendental equations:**		3
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.		
Practical 3	**Interpolation**		3
a.	Program for Newton's forward interpolation.		
b.	Program for Newton's backward interpolation.		
c.	Program for Lagrange's interpolation.		
Practical 4	**Solving linear system of equations by iterative methods**		3
a.	Program for solving linear system of equations using Gauss Jordan method.		
b.	Program for solving linear system of equations using Gauss Seidel method.		
Practical 5	**Numerical Differentiation**		3
a.	Programing to obtain derivatives numerically.		
Practical 6	**Numerical Integration**		3
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.		
Practical 7	**Solution of differential equations**		3
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.	
Practical 8	**Regression**	**3**
a.	Program for Linear regression.	
b.	Program for Polynomial Regression.	
c.	Program for multiple linear regression.	
d.	Program for non-linear regression.	
Practical 9	**Random variables and distributions**	**3**
a.	Program to generate random variables.	
b.	Program to fit binomial distribution.	
c.	Program to fit Poisson distribution.	
Practical 10	**Distributions**	**3**
a.	Program for Uniform distribution.	
b.	Program for Bernoulli distribution	
c.	Program for Negative binomial distribution.	

Course Code BNBUSIT2P5	Course Title Green Computing Practical	Credits 2	No. of lectures
Practical 1	A project should be done based on the objectives of Green Computing. A report of minimum 50 pages should be prepared. The report should have a font size of 12, Times new roman and 1.5 line spacing. The headings should have font size 14. The report should be hard bound.		3
Practical 2	The project can be done individually or a group of two students.		3
Practical 3	The students will have to present the project during the examination.		3
Practical 4	A certified copy of the project report is essential to appear for the examination.		3

References

Course Code	Course Title
BNBUSIT2T1	**Object Oriented Programming**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
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
3.	C++ for beginners	B. M. Hirwani	SPD		2013
4.	Effective Modern C++	Scott Meyers	SPD		
5.	Object Oriented Programming with C++	E. Balagurusamy	Tata McGraw Hill	4th	
6.	Learning Python	Mark Lutz	O' Reilly	5th	2013
7.	Mastering Object Oriented Python	Steven F. Lott	Pact Publishing		2014

Course Code	Course Title
BNBUSIT2T2	**Microprocessor Architecture**

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Microprocessors Architecture, Programming and Applications with the 8085.	Ramesh Gaonkar	PENRAM	Fifth	2012
2.	Computer System Architecture	M. Morris Mano	PHI		1998
3.	Structured Computer Organization	Andrew C.Tanenbaum	PHI		

Course Code	Course Title
BNBUSIT2T3	**Web Programming**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Web Design The Complete Reference	Thomas Powell	Tata McGraw Hill		-
2.	HTML5 Step by Step	Faithe Wempen	Microsoft Press		2011
3.	PHP 5.1 for Beginners	Ivan Bayross Sharanam Shah,	SPD		2013
4.	PHP Project for Beginners	SharanamShah, Vaishali Shah	SPD		2015

Sr. No.	Title	Author/s	Publisher	Edition	Year
5.	PHP 6 and MySQL Bible	Steve Suehring, Tim Converse, Joyce Park	Wiley		2009
6.	Head First HTML 5 Programming	Eric Freeman	O'Reilly		2013
7.	JavaScript 2.0: The Complete Reference	Thomas Powell and Fritz Schneider	Tata McGraw Hill	2nd	

Course Code	Course Title
BNBUSIT2T4	**Numerical and Statistical Methods**

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Introductory Methods of Numerical Methods	S. S. Shastri	PHI	Vol – 2	
2.	Numerical Methods for Engineers	Steven C. Chapra, Raymond P. Canale	Tata Mc Graw Hill	6th	2010
3.	Numerical Analysis	Richard L. Burden, J. Douglas Faires	Cengage Learning	9th	2011
4.	Fundamentals of Mathematical Statistics	S. C. Gupta, V. K. Kapoor			
5.	Elements of Applied Mathematics	P.N.Wartikar and J.N.Wartikar	A. V. Griha, Pune	Volume 1 and 2	

Course Code	Course Title
BNBUSIT2T5	**Green Computing**

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Green IT	Toby Velte, Anthony Velte, Robert Elsenpeter	McGraw Hill		2008
2.	Green Data Center: Steps for the Journey	Alvin Galea, Michael Schaefer, Mike Ebbers	Shroff Publishers		2011
3.	Green Computing and Green IT Best Practice	Jason Harris	Emereo		
4.	Green Computing Tools and Techniques for Saving Energy, Money and Resources	Bud E. Smith	CRC Press		2014

Evaluation Scheme

Internals

Class Test	Active Participation & Leadership qualities	Total
30		
Certification of Swayam / NPTEL in concern course	**10**	**40**

Internal Examination :   Based on Unit 1 / Unit 2 / Unit 3

Duration : 1 Hour   **Total Marks : 20**

	Answer the following	**10**
Q. 1		
Q. 2		
Q. 3		
Q. 4		
Q. 5		

Theory Examination :   Suggested Format of Question paper

Duration : 2 Hours   **Total Marks : 60**

- **All questions are compulsory**

Q. 1		Answer *any two* of the following	**16**
	a	Based on Unit I	
	b	Based on Unit I	
	c	Based on Unit I	
	d	Based on Unit I	
Q. 2		Answer *any two* of the following	**16**
	a	Based on Unit II	
	b	Based on Unit II	
	c	Based on Unit II	
	d	Based on Unit II	
Q. 3		Answer *any two* of the following	**16**
	a	Based on Unit III	
	b	Based on Unit III	
	c	Based on Unit III	
	d	Based on Unit III	
Q. 4		Answer *any two* of the following	**12**
	a	Based on Unit I	
	b	Based on Unit II	
	c	Based on Unit III	

** (4 questions of 8 marks each / 8 questions of 4 marks can be asked)

Practical Examination

F.Y.B.Sc. (I.T.) Semester I/II Practical Examination "Month & Year "

Paper Code :- _____

Duration :- 02.30 hrs.   Total Marks :- 50

Q. No	Question's	Marks
Q. 1.	Program 1.	10
Q. 2.	Program 2.	10
Q. 3.	Program 3.	20
Q. 4.	Viva-Voce	5
Q. 5.	Journal	5

Marks Distribution and Passing Criterion for Each Semester

Theory					Practical		
Course Code	Internal	Min marks for passing	Theory Examination	Min marks for passing	Course Code	Practical Examination	Min marks for passing
BNBUSIT1T1	**40**	**16**	**60**	24	BNBUSIT1P1	**50**	**20**
BNBUSIT1T2	**40**	**16**	**60**	24	BNBUSIT1P2	**50**	**20**
BNBUSIT1T3	**40**	**16**	**60**	24	BNBUSIT1P3	**50**	**20**
BNBUSIT1T4	**40**	**16**	**60**	24	BNBUSIT1P4	**50**	**20**
BNBUSIT1T5	**40**	**16**	**60**	24	BNBUSIT1P5	**50**	**20**

Theory					Practical		
Course Code	Internal	Min marks for passing	Theory Examination	Min marks for passing	Course Code	Practical Examination	Min marks for passing
BNBUSIT2T1	**40**	**16**	**60**	24	BNBUSIT2P1	**50**	**20**
BNBUSIT2T2	**40**	**16**	**60**	24	BNBUSIT2P2	**50**	**20**
BNBUSIT2T3	**40**	**16**	**60**	24	BNBUSIT2P3	**50**	**20**
BNBUSIT2T4	**40**	**16**	**60**	24	BNBUSIT2P4	**50**	**20**
BNBUSIT2T5	**40**	**16**	**60**	24	BNBUSIT2P5	**50**	**20**

~ * ~ * ~ * ~ * ~ * ~