Karmaveer Bhaurao Patil College, Vashi
Autonomous College
Affiliated to University of Mumbai
Syllabus for Approval

Sr. No.	Heading	Particulars
1	Title of Course	Computer Science
2	Eligibility for Admission	3 KT's per SEM
3	Passing marks	40
4	Ordinances/Regulations (if any)	
5	No. of Semesters	Two
6	Level	U.G.
7	Pattern	Choice Based Credit System
8	Status	Revised
9	To be implemented from Academic year	2021-22

AC -

Item No-

Rayat Shikshan Sanstha's
KARMAVEER BHAURAO PATIL COLLEGE, VASHI.
NAVI MUMBAI
(AUTONOMOUS COLLEGE)

Sector-15- A, Vashi, Navi Mumbai - 400 703

Syllabus for F.Y.B.Sc. Computer Science
Program: B.Sc. Computer Science (Honours)
Course: F.Y.B.Sc. Computer Science

(Choice Based Credit System with effect from the academic year 2021‑22)

Preamble

The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skills sets demanded under new technological environment. The proposed curriculum is more contextual, industry affable and suitable to cater the needs of society and nation in present day context.

Second year of this course is about studying core computer science subjects. Theory of Computation course provides an understanding of grammar, syntax and other elements of modern language designs. It also covers developing capabilities to design formulations of computing models and its applications in diverse areas.
The course in Linux Operating System satisfies the need of understanding the structure and functioning of system. Programming holds key indispensable position in any curriculum of Computer Science. It is essential for the learners to know how to use object oriented paradigms(JAVA). There is also one dedicated course Android Developer Fundamentals as a skill enhancement catering to modern day needs of Mobile platforms and applications. The syllabus has Database Systems courses in previous semesters. The course in Database Management Systems is its continuation in third semester.

The course of Combinatorics and Graph Theory in third semester and the course of Linear Algebra in fourth semester take the previous courses in Mathematics. Graph theory is rapidly moving into the mainstream mainly because of its applications in diverse fields which include can further open new opportunities in the areas of genomics, communications networks and coding theory, algorithms and computations and operations research.

Introducing one of the upcoming concepts Physical Computing and IoT programming will definitely open future area as Embedded Engineer, involvement in IoT projects, Robotics and many more. The Raspberry Pi is a popular platform as it offers a complete Linux server in a tiny platform for a very low cost and custom-built hardware with minimum complex hardware builds which is easier for projects in education domain.

Scheme of examination for Each Semester:

Continuous Internal Evaluation: 40 Marks (Common Test-20 Marks & 20 Marks for- Mini Projects, Presentation, Online Course, Case Study, Assignment, Analysis In Statistics, Report Writing, Interviews, Paper Review, Surprise Test).

Semester End Examination: 60 Marks will be as follows-

I.	Theory: The Semester End Examination for theory course work will be conducted as per the following scheme.		
	Each theory paper shall be of two hours duration.		
	All questions are compulsory and will have internal options.		
	Q – I	From Unit – I (having internal options.) 15 M	
	Q – II	From Unit – II (having internal options.)15 M	
	Q – III	From Unit – III (having internal options.)15 M	
	Q – IV	Questions from all the THREE Units with equal weightage of marks allotted to each Unit. 15 M	
II.	Practical	The Semester End Examination for practical courses will be conducted as per the following scheme.	
Sr. No.	**Particulars of Semester End Practical Examination**		**Marks**
1	Laboratory Work		40
2	Journal		05
3	Viva		05
	TOTAL		**50**

F.Y.B.Sc. Computer Science Syllabus

Credit Based System and Grading System

Academic year 2021-2022

SEMESTER - I

CODE	COURSE TYPE	SUBJECT	SCHEME OF INSTRUCTION (PERIOD PER WEEK)		SCHEME OF EXAMINATION (MAX MARKS)			NO. OF CREDITS
			TH	LAB	CIA	SEE	TOTAL	
UGCS101	Core	Computer Organization Architecture	3	-	40	60	100	3
UGCSP101	Core Subject Practical	UGCS101 Practical	-	2			50	1
UGCS102	Core	Python - I	3	-	40	60	100	3
UGCSP102	Core Subject Practical	UGCS102 Practical	-	2			50	1
UGCS103	Core	C Programming	3	-	40	60	100	3
UGCSP103	Core Subject Practical	UGCS103 Practical	-	2			50	1
UGCS104	Ability Enhancement	Soft Skill Development	3	-	40	60	100	3
UGCSP104	Ability Enhancement Practical	UGCS104 Practical	-	2			50	1
UGCS105	Generic Elective - I	Discrete Mathematics	4	-	40	60	100	4
UGCSP105	Generic Elective Practical	UGCS105 Practical	-	4			50	2
UGCS106	Generic Elective - I	Digital Electronics	4	-	40	60	100	4
UGCSP106	Generic Elective Practical	UGCS106 Practical	-	4			50	2
						TOTAL	750	22

SEMESTER - II

CODE	COURSE TYPE	SUBJECT	SCHEME OF INSTRUCTION (PERIOD PER WEEK)		SCHEME OF EXAMINATION (MAX MARKS)			NO. OF CREDITS
			TH	LAB	CIA	SEE	TOTAL	
UGCS201	Core	Database Management systems	3	-	40	60	100	3
UGCSP201	Core Subject Practical	UGCS201 Practical	-	2			50	1
UGCS202	Core	Python - II	3	-	40	60	100	3
UGCSP202	Core Subject Practical	UGCS202 Practical	-	2			50	1
UGCS203	Core	Data Structure using Python	3	-	40	60	100	3
UGCSP203	Core Subject Practical	UGCS203 Practical	-	2			50	1
UGCS204	Ability Enhancement	Free & Open Source Software	3	-	40	60	100	3
UGCSP204	Ability Enhancement Practical	UGCS204 Practical	-	2			50	1
UGCS205	Generic Elective - I	Descriptive Statistics - I	4	-	40	60	100	4
UGCSP205	Generic Elective Practical	UGCS205 Practical	-	4			50	2
UGCS206	Generic Elective - I	Physical Computing and IoT Programming	4	-	40	60	100	4
UGCSP206	Generic Elective Practical	UGCS206 Practical	-	4			50	2
						TOTAL	750	22

SEMESTER I

Class: F.Y.B.Sc	**Branch:** Computer Science	**Semester:** I	
Subject: Computer Organization Architecture			
Period per Week(Each 48 min)	Lecture	03	
	Practical	02	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	2 hrs	50
	Total	—	150

Course: UGCS101	**Computer Organization Architecture** (Credits : 3 Lectures/Week: 3)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students would be able to: 1. Recall the history and development of modern computers. 2. Discuss the number systems and its interconversion. 3. Interpret the concept of memory organization. 4. Analyze the 8085 microprocessor & 8051 microcontroller.	
Unit I	**Introduction:** **a) Computers:** History of computers, and their classification . **b) Memory:** Primary Memory – RAM, SRAM, DRAM, ROM, EPROM. Secondary Memory – Magnetic Floppy and Hard Disk. Optical Memory – CDROM, WORM, Concept of Virtual Memory, Concept of Cache, Memory hierarchy. **c) Number Systems:** Binary, Decimal, Octal, Hexadecimal and their interconversions. **d) Computer Arithmetic:** Binary addition and subtraction using signedMagnitude, 1's complement and 2's complement, Binary multiplication and division,Floating point representation and arithmetic, arithmetic through stacks. **e) Codes for character representation:** hexadecimal, BCD, Excess-3, Gray code, ASCII, EBCDIC, Unicode.	**15l**
Unit II	**Digital logic circuits:** **a) Logic Gates:** AND, OR, NOT, NOR, NAND, EX-OR EX-NOR operations and their truth table, Minimization of gates by K-maps.	**15l**

	b) Digital Circuits: Half Adder, Full Adder, Binary adder-subtractor, binary incrementer, Multiplexers, Encoder and decoder. **c)Flip Flops:** Concept of sequential circuits, concept of clock and synchronization, S-R, J-K, Preset and Clear, Master-Slave J-K, D, T Flip Flops, their truth tables and identities, concept of counters and registers, shift registers.	
Unit III	**a)Input/output devices:** Input/output devices, input/output interface, asynchronous data transfer, modes of data transfer, I/O modules, Concepts of programmed I/O, interrupt Drive I/O, DMA, I/O processors **b)CPU:** Functions of CPU, register classification and organization, instruction sets and examples of instruction set, addressing schemes, instruction formats, instruction cycle and instruction pipelining **Introduction to 8085 architecture:**Overview, Pin Diagram,Block Diagram	**15l**
Unit IV	**8051 Microcontroller** 8 bit Microcontroller architecture, features, advantages and applications, Definition of embedded system and its characteristics. Role of microcontroller in embedded System. Limitation of 8 bit microcontrollers.	**15l**

Textbook(s):

1. "Computer Organization and Architecture: Designing for Performance" by William Stallings
2. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010
3. M. Moris Mano, "Computer Systems Architecture", 4th Edition, Pearson/PHI, ISBN:10:0131755633

Additional Reference(s):

1. http://ac.aua.am/arm/public/2017-Spring-Computer-Organization/Textbooks/ComputerOrganizationAndDesign5thEdition2014.pdf
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5 th Edition, McGraw Hill
3. A.Anandkumar, "Fundamentals of digital circuits", 4th edition, PHI.

Links:

1. https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorials/
2. https://en.wikibooks.org/wiki/IB/Group_4/Computer_Science/Computer_Organisatio
3. https://www.tutorialspoint.com/microprocessor/microcontrollers_8051_architecture.htm
4. https://www.electronicshub.org/8051-microcontroller-introduction/

Sr. No.	Practical's of UGCSP101
1	Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR).
2	Simplify given Boolean expression and realize it.
3	Design and verify a half/full adder.
4	Design and verify half subtractor.

5	Design and verify full subtractor.
6	Design a 4 bit magnitude comparator using combinational circuits.
7	Design and verify the operation of flip-flops using logic gates.
8	Verify the operation of a counter.
9	Verify the operation of a 4 bit shift register
10	Design and verify master slave(JK flip flop) operation and d flip flop.

Class: F.Y.B.Sc	**Branch**: Computer Science	**Semester:** I	
Subject: Python - I			
Period per Week(Each 60 min)	Lecture	03	
	Practical	02	
Evaluation System		Hours	Marks
	Semester End Exam	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	2 hrs	50
	Total	__	150

Course Code	**Course Title**	**Credits**	**Lectures /Week**
UGCS102	**Python I**	**3**	**3**

Expected Learning Outcomes:
After successful completion of this course, students would be able to:

1. Define logic statements.
2. Identify and apply various properties relating to the integers.
3. Apply different methods of proof to verify mathematical assertions.
4. Apply Fundamental theorem of algebra for finding roots of given polynomials.

Unit	**Topics**	**No of Lectures**

I	**Overview of Python:** History & Versions, Features of Python, Execution of a Python Program, Flavours of Python, Innards of Python, Python Interpreter, Memory Management in Python, Garbage Collection in Python, Comparison of Python with C and Java, Installing Python, Writing and Executing First Python Program, Getting Help, IDLE **Data Types, Variables and Other Basic Elements:** Comments, Docstrings, Data types- Numeric Data type, Compound Data Type, Boolean Data type, Dictionary, Sets, Mapping, Basic Elements of Python, Variables **Input and Output Operations:** Input Function, Output Statements, The print() function, The print("string") function, The print(variables list) function, , The print(object) function, The print(formatted string) function, Command Line Arguments **Operators:** Arithmetic operators, Assignment operators, Unary minus operator, Relational operators, Logical operators, Bitwise operators, Membership operators, Identity operators, Precedence of Operators, Associativity of Operators	15
II	**Control Statements:** The if statement, The if … else Statement, The 'if … elif … else' Statement, Loop Statement- while loop, for loop, Infinite loop, Nested loop, The else suite, break statement, continue statement, pass statement, assert statement, return statement **Functions:** Function definition and call, Returning Results, Returning Multiple Values from a Function, Built-in Functions, Difference between a Function and a Method, Pass Value by Object Reference, Parameters and Arguments, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Arbitrary Arguments, Recursive Functions, Anonymous or Lambda Functions, Using Lambda with the filter() Function, Using Lambda with the map() Function, Using Lambda with the reduce() Function **Modules:**Introduction to Modules in Python	15
III	**Strings:** Creating Strings, Functions of Strings, Working with Strings, Length of a String, Indexing and Slicing, Repeating and Concatenating Strings, Checking Membership, Comparing Strings, Removing Spaces, Finding Substrings, Counting Substrings, Immutability, Splitting and Joining Strings, Changing Case, Checking Starting and Ending of a String, Sorting Strings, Searching in the Strings, Testing Methods, Formatting Strings, Finding the Number of Characters and Words, Inserting Substrings into a String **List and Tuples:** Lists, List Functions and Methods, List Operations, List Slices, Nested Lists, Tuples, Functions in Tuple **Dictionaries:** Creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries, Converting Lists into Dictionary, ConvertingStrings into Dictionary,	15

	Passing Dictionaries to Functions, Sorting the Elements of a Dictionary using Lambda, Ordered Dictionaries	
IV	**Working with files:** Files, opening and closing a file, working with text files containing strings, knowing whether a file exists or not, working with binary files, the 'with' statement, the seek() and tell() methods, random accessing of binary files, zipping and unzipping files, working with directories, running other programs from python program **Regular expressions:** What is a regular expression?, sequence characters in regular expressions, quantifiers in regular expressions, special characters in regular expressions, using regular expressions on files, retrieving information from an html file.	**15**

Textbooks:

1. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries , Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition, 2014
2. Programming through Python, M. T Savaliya, R. K. Maurya& G M Magar, Sybgen Learning India, 2020

Additional References:

1. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018
2. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017
3. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018
4. Python Programming: Using Problem Solving Approach, ReemaThareja, Oxford Univeristy Press, 2017
5. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019

Course Code	**Course Title**	**Credits**	**Lectures/Week**
UGCSP102	**Python – I**	**1**	**2**
1	Write a program to design and develop python program to implement various control statement using suitable examples		
2	Write program in Python to define and call functions for suitable problem.		
3	Write Python program to demonstrate different types of function arguments.		
4	Write a Python program to demonstrate the precedence and associativity of operators.		

5	Write suitable Python program to implement recursion for problems such as Fibonacci series, Factorial, Tower of Hanoi etc.
6	Write Python program to implement and use lambda function in python
7	Write a python program to create and manipulate arrays in Python. Also demonstrate use of slicing and indexing for accessing elements from the array.
8	Write a program to implement list in Python for suitable problem. Demonstrate various operations on it.
9	Write a program to implement tuple in Python for suitable problem. Demonstrate various operations on it.
10	Write a program to implement dictionary in Python for suitable problem. Demonstrate various operations on it.
11	Write a program to Python program to implement various file operations.
12	Write a program to Python program to demonstrate use of regular expression for suitable application.

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** I	
Subject : C Programming			
Period per Week(Each 60 min)	Lecture	03	
	Practical	02	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	2hrs	50
	Total	—	150

Course: PGCS103	C Programming (Credits : 3 Lectures/Week: 3)	Lectures
	Expected Course Outcomes: After successful completion of this course, students will be able to: 1. Explain the programming environment with C Program structure. 2. Recognize the different data types, variables, operators in C programming. 3. Design programs involving decision structures, loops and functions. 4. Describe the array , string & use of pointers,structures, union & dynamic memory allocation in C.	
Unit I	**Introduction to Programming Languages :** What is program & programming paradigms, Programming languages-their classification and characteristics, language translators and language translation activities, Use of Algorithms/Flow Charts for problem solving **Structure of C program**: Header and body, Use of comments. Interpreters vs compilers, Compilation of a program. Formatted I/O: printf(), scanf(). **Data**: Variables, Constants, data types like: int, float char, double and void, short and long size qualifiers, signed and unsigned qualifiers.	**15l**
Unit II	**Variables**: Declaring variables, scope of the variables according to block, hierarchy of data types. **Types of operators**: Arithmetic, relational, logical, compound assignment, increment and decrement, conditional or ternary, bitwise and comma operators. Precedence and order of evaluation, statements and Expressions. Automatic and explicit type conversion. **Iterations**: Control statements for decision making: (i) Branching: if statement, else.. if statement, (does the writer mean if-else or nested ifs)switch statement. (ii) Looping: while loop, do.. while, for loop. (iii) Jump statements: break, continue and goto.	**15l**
Unit III	**Arrays:** (One and two dimensional), declaring array variables, initialization of arrays, accessing array elements. **Data Input and Output functions**: Character I/O format: getch(), getche(), getchar(),getc(), gets(), putchar(), putc(), puts(). **Functions**: Function declaration, function definition, Global and local variables, return statement, Calling a function by passing values,Recursive functions. **Manipulating Strings**: Declaring and initializing String variables, Character and string handling functions.	**15l**
Unit IV	**Pointer:** Fundamentals, Pointer variables, Referencing and dereferencing, Pointer Arithmetic. **Dynamic Memory Allocation**: malloc(), calloc(), realloc(), free() and sizeof operator. **Structure**: Declaration of structure, reading and assignment of structure variables. **Unions**: Defining and working with unions.	**15l**

	Text books: 1) Programming with C by Byron Gottfried Tata McGRAWHill 2) Let us C Yashwant P. Kanetkar, BPB publication 3) C for beginners Madhusudan Mothe X-Team 4) Ansi C Balaguruswamy-c Language **Additional References:** 1) The C Programming Language (2nd Edition) - Brian W. Kernighan and Dennis M. Ritchie (1988) 2) Programming in C (4th Edition) - Stephen Kochan (2014). **Links:** 1) https://www.javatpoint.com/c-programming-language-tutorial 2) https://www.programiz.com/c-programming 3) https://www.geeksforgeeks.org/c-language-set-1-introduction/ 4) https://en.wikibooks.org/wiki/C_Programming	

Sr.No	**Practicals of PGCSP103**
1	Programs to understand the basic data types and I/O.
2	Programs on Operators and Expressions
3	Programs on decision statements.
4	Programs on looping.
5	Programs on arrays.
6	Programs on functions.
7	Programs on structures and unions.
8	Programs on pointers.
9	Programs on string manipulations.
10	Programs on basic file operations.
Note	Programs to understand the basic data types and I/O.

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** I	
Subject : Soft Skill Development			
Period per Week(Each 60 min)	Lecture	03	
	Practical	02	
Evaluation System		**Hours**	**Marks**

	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	2hrs	50
	Total	__	150

Course: UGCS104	**Soft Skill Development** **(Credits : 3 Lectures/Week: 3)**	**Lectures**
	Expected Course Outcomes: After successful completion of this course, students will be able to: 1. Discuss the various aspects of soft skills and learning ways to develop personality. 2. Recognize one's self learning, emotional handling,Etiquette, Mannerism and Academic Skills. 3. Interpret today's communication ,multitasking with time,Public Speaking. 4. Illustrate Professional Skills,Leadership,Decision Making,Stress and Time Management.	
Unit I	**Introduction to Soft Skills and Hard Skills** **Personality Development:** Knowing Yourself, Positive Thinking, Johari's Window, Communication Skills, Non-verbal Communication, Physical Fitness **Emotional Intelligence:** Meaning and Definition, Need for Emotional Intelligence, Intelligence Quotient versus Emotional Intelligence Quotient, Components of Emotional Intelligence Competencies of Emotional Intelligence, Skills to Develop Emotional Intelligence. **Etiquette and Mannerism:** Introduction, Professional Etiquette, Technology Etiquette ories of Ethics, Correlation between Values and Behavior, Nurturing Ethics, Importance of Work Ethics, Problems in the Absence of Work Ethics	**15l**
Unit II	**Communication Today:** Significance of Communication, GSC's 3M Model of Communication, Vitality of the Communication Process, Virtues of Listening, Fundamentals of Good Listening, Nature of Non-Verbal Communication, Need for Intercultural Communication, Communicating Digital World. **Multi-Tasking** How to prioritize the work, Importance of multi-tasking and concerns related to multitasking, To identify what to multi-task. **Public Speaking** To get acquainted with the art of public speaking, To know the rhetoric of making a public speech, Exploring rhetorical elements through various videos.	**15l**
Unit III	**Academic Skills** **Employment Communication:** Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter **Professional Presentation:** Nature of Oral Presentation, Planning a Presentation, Preparing the Presentation, Delivering the Presentation	**15l**

	Job Interviews: Introduction, Importance of Resume, Definition of Interview, Background Information, Types of Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, Changes in the Interview Process, FAQ During Interviews **Group Discussion:** Introduction, Ambience/Seating Arrangement for Group Discussion, Importance of Group Discussions, Difference between Group Discussion, Panel Discussion and Debate, Traits, Types of Group Discussions, topic based and Case based Group Discussion, Individual Traits	
Unit IV	**Professional Skills Creativity at Workplace:** Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method **Leadership and Team Building:** Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams. **Decision Making and Negotiation:** Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts**.** **Stress and Time Management:** Stress, Sources of Stress*,* Ways to Cope with Stress	**15l**
	Text Books: 1) Personality Development and Soft Skills, Barun K. Mitra, Oxford Press 2) Business Communication, Shalini Kalia, Shailja Agrawal, Wiley India 3) Soft Skills: Personality Development For Life Success Paperback – Prashant Sharma **Additional References:** 1) Soft Skills - Enhancing Employability, M. S. Rao, I. K. International 2) Cornerstone: Developing Soft Skills, Sherfield, Pearson India **Links:** 1) https://en.wikipedia.org/wiki/Soft_skills 2) http://www.ijqr.net/journal/v10-n1/5.pdf 3) https://www.indeed.com/career-advice/career-development/how-to-improve-soft-skills	

Sr. No	Practicals of UGCSP104
1	Presentation For Verbal Communication.
2	Study the Email Etiquettes .
3	Create Report Writing Drafting.
4	Understand the Situation Based Multitasking.
5	CV Writing, Resume Writing & Cover Letter.
6	Arrange the Debate on two particular topics .

7	Arrange the Group Discussion.
8	Implement the various Creativity Activities.
9	Implement the various Time Management Activities.
10	Mock Interviews

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** I	
Subject : Discrete Mathematics			
Period per Week(Each 60 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	2hrs	50
	Total	—	150

Course: UGCS105	**Discrete Mathematics (Credits : 4 Lectures/Week: 4)**	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students would be able to: 1. Recall the concepts of set, function and logic. 2. Describe recurrence relations & evaluate recurrence relations using different methods. 3. Develop the knowledge of permutations and combinations and counting principles. 4. Analyze the basic concepts of graphs and trees, languages, regular expressions ,problems on finite state automata and Turing Machine.	
Unit I	**Set Theory :** Fundamentals - Sets and subsets, Venn Diagrams, Operations on sets, Laws of Set Theory, Power Sets and Products, Partition of sets, The Principle of Inclusion - Exclusion. **Logic :** Propositions and Logical operations, Truth tables, Equivalence, Implications, Laws of Logic, Normal forms, Predicates and quantifiers, Mathematical Induction. **Functions:** Definition of function. Domain, co domain and the range of a function. Direct and inverse images. Injective, surjective and bijective functions. Composite and inverse functions.	**15l**

Unit II	**Relations**: Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Daigrams **Recurrence Relations:** Definition of recurrence relations, Formulating recurrence relations **Methods of solving recurrence relations-** Back tracking method, Linear homogeneous recurrence relation with constant coefficients. **Permutations and Combinations:** Partition and Distribution of objects, Permutation with distinct and indistinct objects, Combination with indistinct objects	**15l**
Unit III	**Counting Principles:** Sum and Product Rules, Two-way counting, Tree diagram for solving counting problems, Pigeonhole Principle (without proof); Simple examples, **Graphs :** Definition and elementary results, Adjacency matrix, path matrix, Representing relations using digraphs. **Trees:** Definition and elementary results. Ordered rooted tree, Binary trees, Complete and extended binary trees	**15l**
Unit IV	**Languages**: Languages,Formal & Natural language,Operation on languages, regular Expression and Regular languages, **Grammars :** Regular grammars, closures properties, Gödel numbers **Introduction to Machines:** Finite state Automata, Finite state machines, Introduction to NFA & DFA, Turing machines	**15l**
	Textbooks: 1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011) 2. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989. 3. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw-Hill Inc. 4. Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI,3 rd Edition **Additional Reference:** 1. Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition . 2. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education. 3. Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc. 4. Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New Delhi. Theory of Computation, Kavi Mahesh, Wiley India 5. Introduction to Languages and the Theory of Computation, John 6. E Martin, McGraw-Hill Education **Links:** 1. https://www.tutorialspoint.com/discrete_mathematics/	

	2. http://www.cs.yale.edu/homes/aspnes/classes/202/notes.pdf 3. https://www2.cs.duke.edu/courses/spring09/cps102/Lectures/Book.pdf 4. https://mcdtu.files.wordpress.com/2017/03/toc-klp-mishra.pdf 5. https://pakistandasti.files.wordpress.com/2013/11/introduction-to-computer-theory-by-cohen-copy.pdf	

Sr.No.	**Practicals of UGCSP105**
1	Set Theory - a. Laws of Set Theory b. The Principle of Inclusion - Exclusion Logic - a. Propositions and Logical operations b. Equivalence Mathematical Induction
2	Functions – a. Identify if the given mapping is a function b. Finding domain and range of a given function c. Check if the given function is injective/surjective/bijective d. Find the inverse of a given function e. Graphs of functions using any online tool
3	Relations – a. Representation of relations b. Determine if the given relation satisfies equivalence relation/partial order relation c. Finding transitive closure using Warshall's Algorithm d. Draw Hasse diagrams
4	Recurrence Relation – a. Solve recurrence relation using backtracking method b. Solve linear homogeneous recurrence relations with constant coefficients c. Find homogeneous, particular, general solution of a recurrence relation d. Formulate and solving recurrence relation
5	Counting Principles – a. Sum and product rule b. Pigeonhole Principle c. Inclusion Exclusion Principle d. Counting using Tree diagrams
6	Permutations and Combinations – a. Permutations b. Permutations with repetitions c. Combinations

	d. Combinations with repetitions e. Binomial numbers and Identities f. Applications on Binomial theorem g. Applications on Multinomial theorem
7	Graphs – a. Types of graph b. Properties of graph c. Representation of graph
8	Trees – a. Tree terminologies b. Types of tree c. Properties of tree d. Representation of tree e. Expression tree
9	Finite State Machines – a. Check if a given string is accepted or rejected by FSM without output b. Find the output for a FSM with output c. Describe a machine (diagram/table)
10	Regular Expression and Regular Language – a. Describe the regular expressions represented by given language b. Describe the language represented by given regular expression

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** I	
Subject : Digital Electronics			
Period per Week(Each 60 min)	Lecture	04	
	Practical	04	
Evaluation System	.	**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	2 hrs	50
	Total	__	150

Course: UGCS106	**Digital Electronics** **(Credits : 4 Lectures/Week: 2)**	**Lecture**
	Expected Learning Outcomes: After successful completion of this course, students will be able to:	

	1. Recognize the knowledge of computer systems work and underlying principles and the basics of digital electronics needed for computers. 2. Recall the basics of instruction set architecture for reduced and complex instruction sets 3. Describe the basics of processor structure and operation 4. Explain how data is transferred between the processor and I/O devices	
Unit I	**Fundamentals of Digital Logic:** Boolean algebra, Logic Gates, Simplification of Logic Circuits: Algebraic Simplification, Karnaugh Maps. Combinational Circuits: Adders, Mux, De-Mux, Sequential Circuits: FlipFlops (SR, JK & D), Counters: synchronous and asynchronous Counter. **Computer System**: Comparison of Computer Organization & Architecture, Computer Components and Functions, Interconnection Structures. Bus Interconnections, Input / Output: I/O Module, Programmed I/O, Interrupt Driven I/O, Direct Memory Access .	**15l**
Unit II	**Memory System Organization:** Classification and design parameters, Memory Hierarchy, Internal Memory: RAM, SRAM and DRAM, Interleaved and Associative Memory. Cache Memory: Design Principles, Memory mappings, Replacement Algorithms, Cache performance, Cache Coherence. Virtual Memory, External Memory: Magnetic Discs, Optical Memory, Flash Memories, RAID Levels .	**15l**
Unit III	**Processor Organization:** Instruction Formats, Instruction Sets, Addressing Modes, Addressing Modes Examples with Assembly Language [8085/8086 CPU], Processor Organization, Structure and Function. Register Organization, Basic Microprocessor operations: Data Transfer (Register / Memory) Operations, Arithmetic & Logical Operations, Instruction Cycle, Instruction Pipelining.	**15l**
Unit IV	**Control Unit:** Micro-Operations, Functional Requirements, Processor Control, Hardwired Implementation, Micro-programmed Control. **Fundamentals of Advanced Computer Architecture:** Parallel Architecture: Classification of Parallel Systems, Flynn's Taxonomy, Array Processors, Clusters, and NUMA Computers. Multiprocessor Systems: Structure & Interconnection Networks, Multi-Core Computers: Introduction, Organization and Performance.	**15l**
	TEXT BOOKS: 1. M. Mano, Computer System Architecture 3rd edition, Pearson 2. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012 3. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010 **Additional References:** 1. William Stallings (2010), Computer Organization and Architecture-designing for performance,8th edition, Prentice Hall, New Jersy. 2. Anrew S. Tanenbaum (2006), Structured Computer Organization, 5th edition, PearsonEducation Inc, 3. John P. Hayes (1998), Computer Architecture and Organization, 3rd edition, Tata McGrawHill. **Links:**	

	https://www.tutorialspoint.com/digital_circuits/index.htm https://www.studytonight.com/computer-architecture/memory-organization	

Sr.No.	**Practicals of UGCSP106**
1	Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR).
2	Study of Logic gates and their ICs 7400, 7402, 7404, 7408, 7432, 7486, 74266 and universal gates
3	Design and implement 8:3 encoder .
4	Design and implement 3:8 decoder.
5	Design and implement 1:4 demultiplexer. Study of IC 74139
6	Design a 4 bit magnitude comparator using combinational circuits Design and implement combinational circuit based on the problem given and minimizing using K-maps
7	Verifying De Morgan's laws.
8	Implement other given expressions using minimum number of gates.
9	Implement other given expressions using minimum number of ICs.
10	Design serial – in serial – out, serial – in parallel – out, parallel – in serial – out, parallel – in parallel – out and bidirectional shift registers using IC 7474.
Note	Practical 1 – 10 can be performed using any open source simulator (like Logisim) (Download it from https://sourceforge.net/projects/circuit/)

Semester - II

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** II	
Subject : Database Management systems			
Period per Week(Each 60 min)	Lecture	03	
	Practical	02	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	2hrs	50
	Total	—	150

Course: UGCS201	**Database Management systems (Credits : 3 Lectures/Week: 3)**	**Lectures**
	Expected Course Outcomes: After successful completion of this course, students would be able to 1. Analyze database requirements and determine the entities involved in the system and their relationship to one another. 2. Create tables by using DDL commands,perform query using DML commands in MYSQL. 3. Explain Relational data model & Relational Algebra ,functions,join ,subqueries. 4. Describe the normalization,indexes,views and Database Protection.	
Unit I	**Introduction to DBMS –** Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture. **Data models** - Client/Server Architecture, Object Based Logical Model, Record Based Logical Model (relational, hierarchical, network) **Entity Relationship Model** - Entities, attributes, entity sets, relations, relationship sets, Additional constraints (key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER (entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER)	**15l**
Unit II	**ER to Table-** Entity to Table, Relationship to tables with and without key constraints. **DDL Statements** - Creating Databases, Using Databases, datatypes, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables. **DML Statements** – Viewing the structure of a table insert, update, delete, Select all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg, sum), group by clause, having clause	**15l**

Unit III	**Relational data model–** Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint. **Relational Algebra** operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division) Functions – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math **Functions** (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse). **Joining Tables** – inner join, outer join (left outer, right outer, full outer) **Subqueries** – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries	**15l**
Unit IV	**Schema refinement and Normal forms:** Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition **Database Protection:** Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control, Backing Up and Restoring databases **Views** (creating, altering dropping, renaming and manipulating views) **DCL Statements** (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges), Transaction control commands – Commit, Rollback **Index Structures of Files**: Introduction, Primary index, Clustering Index, Multilevel indexes	**15l**
	Textbooks: 1. "Fundamentals of Database System", ElmasriRamez, NavatheShamkant, Pearson Education, Seventh edition, 2017. 2. "Database Management Systems", Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, 2014 . 3. "Murach's MySQL", Joel Murach, 3rd Edition, 3rd Edition, 2019 **References:** 1. "Database System Concepts", Abraham Silberschatz, HenryF.Korth, S.Sudarshan, McGraw Hill, 2017 2. 2. "MySQL: The Complete Reference", VikramVaswani , McGraw Hill, 2017 3. "Learn SQL with MySQL: Retrieve and Manipulate Data Using SQL Commands with Ease", AshwinPajankar, BPB Publications, 2020. **Links:** https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/ https://www.auhd.site/upfiles/elibrary/Azal2020-01-22-12-28-11-76901.pdf	

Sr.No	Practicals of UGCSP201
1	Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.)
2	Perform the following: • Viewing all databases • Creating a Database • Viewing all Tables in a Database • Creating Tables (With and Without Constraints) • Inserting/Updating/Deleting Records in a Table .
3	Perform the following: • Altering a Table • Dropping/Truncating/Renaming Tables • Backing up / Restoring a Database
4	Perform the following: • Simple Queries • Simple Queries with Aggregate functions
5	Queries involving • Date Functions • String Functions • Math Functions
6	Join Queries • Inner Join • Outer Join
7	Subqueries • With IN clause • With EXISTS clause
8	Converting ER Model to Relational Model and apply Normalization on database. (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys and normalization up to 3rd Normal Form).
9	Views : • Creating Views (with and without check option) • Dropping views • Selecting from a view
10	DCL statements: • Granting and revoking permissions • Saving (Commit) and Undoing (rollback)
11	Creating Indexes on data tables.

Class: F.Y.B.Sc	**Branch**: Computer Science	**Semester:** II	
Subject: Python - II			
Period per Week(Each 60 min)	Lecture	03	
	Practical	02	
Evaluation System		Hours	Marks
	Semester End Exam	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	2 hrs	50
	Total	__	150

Course Code	**Course Title**	**Credits**	**Lectures /Week**
UGCS202	**Python-II**	**3**	**3**

Expected Course Outcomes:

After successful completion of this course, students would be able to

1. Describe OOP concepts in Python including Inheritance and Polymorphism
2. Apply files, regular expression and concept of threads for developing efficient program
3. Illustrate exception handling in Python applications for error handling.
4. Recognize the Knowledge of working with databases, designing GUI in Python and implement networking in Python

Unit	**Topics**	**No of Lectures**
I	**Threads in python:** Difference between process and thread, types of threads, benefits of threads, creating threads, single tasking and multitasking, thread synchronization, deadlock in threads, daemon threads **Date and time in python:** Date and time now, combining date and time,formatting dates and times, finding durations using “time delta”, comparing two dates, sorting dates, stopping execution temporarily, knowing the time taken by a program, calendar module	**15**

II	**Exceptions in python:** Errors in a python program, compile & run-time errors, logical error, exceptions-exception handling, types of exceptions, the except block, the assert statement, user-defined exceptions, logging the exceptions **Database in python:** Using SQL with python, retrieving rows from a table, inserting rows into a table, deleting rows from a table, updating rows in a table, creating database tables through python, Exception handling in databases. **Networking:** Protocols,server-client architecture, tcp/ip and udp communication	**15**
III	**Graphical user interface:** Creating a GUI in python, Widget classes, Working with Fonts and Colours, working with Frames, Layout manager, Event handling **OOPs in python:** Features of Object Oriented Programming system (oops)-classes and objects, encapsulation, abstraction, inheritance, polymorphism, constructors and destructors **Classes and objects:** Creating a class, the self-variable, types of variables, namespaces, types of methods, instance methods, class methods, static methods, passing members of one class to another class, inner classes	**15**
IV	**Inheritance and polymorphism:** Inheritance in python, types of inheritance- single inheritance, multilevel inheritance, hierarchical inheritance, multiple inheritance, constructors in inheritance, overriding super class constructors and methods, the super() method, method resolution order (mro), polymorphism, duck typing, operator overloading, method overloading, method overriding**,** **Abstract classes and interfaces:** Abstract class, abstract method, interfaces in python, abstract classes vs. Interfaces	**15**

Textbooks:

1. Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 3rd Edition, 2018
2. Programming through Python, M. T Savaliya, R. K. Maurya, G M Magar, Revised Edition, Sybgen Learning India, 2020

Additional References:

1. Advanced Python Programming, Dr. Gabriele Lanaro, Quan Nguyen, SakisKasampalis, Packt Publishing, 2019
2. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018
3. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018
4. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017
5. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018

Course Code	Course Title	Credits	Lectures/Week
UGCSP202	**Python-II – Practical**	**1**	**2**
1	Write a Program to demonstrate the concept of threading and multitasking in Python.		
2	Write a Python Program to create an application which uses date and time in Python.		
3	Write a Python Program to demonstrate different types of exception handling.		
4	Write a Python Program to work with databases in Python to perform operations such as a. Connecting to database b. Creating and dropping tables c. Inserting and updating into tables.		
5	Write a Python program to create server-client and exchange basic information		
6	Write a GUI Program in Python to design application that demonstrates a. Different fonts and colors b. Different Layout Managers		
7	Write a GUI Program in Python to design application that demonstrates on various widgets		
8	Write a GUI Program in Python to design application that demonstrates on Event Handling		
9	Write a program to Python program to implement concepts of OOP such as a. Types of Methods b. Inheritance c. Polymorphism		
10	Write a program to Python program to implement concepts of OOP such as a. Abstract methods and classes b. Interfaces		

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** II	
Subject : Data Structure using Python			
Period per Week(Each 60 min)	Lecture	03	
	Practical	02	
Evaluation System		**Hours**	**Marks**

	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	2hrs	50
	Total	__	150

Course: UGCS203	**Data Structure using Python (Credits : 3 Lectures/Week: 3)**	**Lectures**
	Expected Course Outcomes: After successful completion of this course, students would be able to 1. Recognize Data structures, its types and significance in computing. 2. Develop searching and sorting techniques 3. Examine the difference between stack and queue 4. Illustrate concepts of binary trees, develop applications using data structure and Evaluate postfix and prefix expressions.	
Unit I	**Abstract Data Types:** Introduction, The Date Abstract Data Type, Bags, Iterators. Application **Arrays**: Array Structure, Python List, Two Dimensional Arrays, Matrix Abstract Data Type, Application **Sets and Maps:** Sets-Set ADT, Selecting Data Structure, List based Implementation, Maps-Map ADT, List Based Implementation, Multi-Dimensional Arrays-Multi-Array ADT, Implementing Multi Arrays, Application **Algorithm Analysis:** Complexity Analysis-Big-O Notation, Evaluating Python Code, Evaluating Python List, Amortized Cost, Evaluating Set ADT, Application	**15l**
Unit II	**Searching and Sorting:** Searching-Linear Search, Binary Search, Sorting-Bubble, Selection and Insertion Sort, Working with Sorted Lists-Maintaining Sorted List, Maintaining sorted Lists **Linked Structures:** Introduction, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, Bag ADT-Linked List Implementation. Comparing Implementations, Linked List Iterators, More Ways to Build Linked Lists, Applications-Polynomials	**15l**
Unit III	**Advanced Linked List:** Doubly Linked Lists-Organization and Operation, Circular Linked List-Organization and Operation, Multi Lists **Stacks:** Stack ADT, Implementing Stacks-Using Python List, Using Linked List, Stack Applications-Balanced Delimiters, Evaluating Postfix Expressions **Queues:** Queue ADT, Implementing Queue-Using Python List, Circular Array, Using List, Priority Queues- Priority Queue ADT, Bounded and unbounded Priority Queues	**15l**
Unit IV	**Recursion:** Recursive Functions, Properties of Recursion, Its working, Recursive Applications **Hash Table:** Introduction, Hashing-Linear Probing, Clustering, Rehashing, Separate Chaining, Hash Functions **Advanced Sorting:** Merge Sort, Quick Sort, Radix Sort, Sorting Linked List	**15l**

	Binary Trees: Tree Structure, Binary Tree-Properties, Implementation and Traversals, Expression Trees, Heaps and Heapsort,Search Trees	
	TextBook: 1) Data Structure and algorithm Using Python, Rance D. Necaise, 2016 Wiley India Edition 2) Data Structure and Algorithm in Python, Michael T. Goodrich, Robertom Tamassia, M. H. Goldwasser, 2016 Wiley India Edition **References:** 1) Data Structure and Algorithmic Thinking with Python-Narasimha Karumanchi, 2015, Careermonk Publications 2) Fundamentals of Python: Data Structures, Kenneth Lambert, Delmar Cengage Learning **Links:** **https://lecturenotes.in/subject/81/data-structure-using-c-ds** **http://www.cs.yale.edu/homes/aspnes/classes/223/notes.pdf** **https://www.smartzworld.com/notes/data-structures-pdf-notes-ds/** **https://www.geeksforgeeks.org/data-structures/**	

Sr.No	Practicals of UGCSP203
1	Implement Linear Search to find an item in a list.
2	Implement binary search to find an item in an ordered list
3	Implement Sorting Algorithms a. Bubble sort b. Insertion sort c. Quick sort d. Merge sort
4	Implement use of Sets and various operations on Sets.
5	Implement working of Stacks. (pop method to take the last item added off the stack and a push method to add an item to the stack)
6	Implement Program for a. Infix to Postfix conversion b. Postfix Evolution
7	Implement the following a. A queue as a list which you add and delete items from. b. A circular queue. (The beginning items of the queue can be reused).
8	Implement Linked list and demonstrate the functionality to add and delete items in the linked list.

9	Implement Binary Tree and its traversals.
10	Recursive implementation of a. Factorial b. Fibonacci c. Tower of Hanoi

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** II	
Subject : Free & Open Source Software			
Period per Week(Each 60 min)	Lecture	03	
	Practical	02	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	2 hrs	50
	Total	—	150

Course: UGCS204	**Free & Open Source Software** **(Credits : 3 Lectures/Week: 2)**	**Lectures**
	Expected Course Outcomes: After successful completion of this course, students would be able to: 1. Discuss knowledge of the Open Source system, its use, impact and importance. 2. Compare between Open source and closed source software. 3. Describe the Life Cycle of an open source project. 4. Compare between the Legal Impacts of Open Software and Free Software Licensing.	
Unit I	**Introduction:** Open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean no cost. **History:** BSD, The Free Software Foundation and the GNU Project. **Methodologies** Open Source History, Initiatives, Principle and methodologies. **Philosophy:** Software Freedom, Open Source Development Model **Licenses and Patents:** What Is A License, Important FOSS Licenses (Apache, BSD, GPL, LGPL), copyrights and copy lefts **Patents Economics of FOSS:** Zero Marginal Cost, Income-generation opportunities, Problems with traditional commercial software, Internationalization	**15l**

Unit II	**Social Impact** Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government. **Examples of Open Source Softwares and Licences**, Open Source Hardware, Open Source Design, Open source Teaching. Open source media. **Collaboration, Community and Communication:** Introduction to github, interacting with the community on github, Introduction to wikipedia.	**15l**
Unit III	**Open Source Development:** Infrastructure Needed for an Open-Source Project, Public code archive, Project documentation, Bug database, Open mailing lists and newsgroup, Project website, Software Development Life Cycle, Building a Community. Joining an Existing Open Source Project, To become a good community member, To adopt development process of OSS, Get the contributions accepted, Ending an Open-Source Project, Open Source Within a Company	**15l**
Unit IV	**Understanding Open Source Ecosystem:** Open Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open Source Hardware, Virtualization Technologies, **Containerization Technologies:** Docker, Development tools, IDEs, debuggers, Programming languages, LAMP, Open Source database technologies **Free Software Licenses.**	**15l**
	Textbooks: 1. The official Ubuntu Book, 8th Edition 2. Understanding Open Source and Free Software Licensing, Andrew M. St. Laurent **References:** 1. For Fun and Profit: A History of the Free and Open Source Software Revolution By Christopher Tozzi 2. Understanding Open Source and Free Software Licensing By Andrew M. St. Laurent 3. Producing Open Source Software How to Run a Successful Free Software Project By Karl Fogel **Links:** 1. The Linux Documentation Project: http://www.tldp.org/ 2. Docker Project Home: http://www.docker.com 3. Linux kernel Home: http://kernel.org 4. Open Source Initiative: https://opensource.org/ 5. Linux Documentation Project: http://www.tldp.org/ 6. Wikipedia: https://en.wikipedia.org/ 7. https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia 8. Github: https://help.github.com/ 9. The Linux Foundation: http://www.linuxfoundation.org/ 10. Open source Development: https://flylib.com/books/en/2.603.1/	

Sr.No.	Practicals of UGCSP204
1	Identify any Open Source software and create detailed report about it. Sample Guidelines. a. Idea b. What problem does it solve? c. Licensing model d. Intent behind making it open source e. Monetization models f. Popularity g. Impact
2	Learn at least three different open source licenses and create a brief report about them. a. History of license b. Idea c. What problems does it solve? d. Detailed licensing model e. Which popular software is released under this license? f. Monetization models g. Any popular news associated with this license? h. Popularity h. Impact
3	Contributing to Open Source a. Identify any Open Source project of your interest b. Learn more about the project w.r.t. Lab 1. c. Start contributing to the project either by i. Testing ii. Reporting bugs iii.Coding iv. Helping in documentation v. Participating in discussions vi. Participating in pre-release testing programs vii. UI development. viii. Or any other important area.
4	Hands on with Open Source Software a. Identify any open source software of your interest b. Learn it from practical view-point c. Give a brief presentation about it to the class d. Sample projects: gcc, gdb, drupal, wordpress, apache web server, mysql database
5	Contributing to Wikipedia: a. Introduction to wikipedia: operating model, license, how to contribute? b. Create your user account on wikipedia c. Identify any topic of your choice and contribute the missing information
6	Github

	a. Create and publish your own open source project: Write any simple program using your choice of programming language. b. Create a repository on github and save versions of your project. You'll learn about the staging area, committing your code, branching, and merging, c. Using GitHub to Collaborate: Get practice using GitHub or other remote repositories to share your changes with others and collaborate on multi-developer projects. You'll learn how to make and review a pull request on GitHub. d. Contribute to a Live Project: Students will publish a repository containing their reflections from the course and submit a pull request.
7	Open Source Operating Systems a. Learn any open source operating system of your choice : Linux, Android, FreeBSD, Open Solaris etc. b. Learn the installation. c. Identify the unique features of the OS of your choice.
8	Virtualization: Open Source virtualization technologies: a. Install and configure any one: VirtualBox, Zen, KVM b. Create and use virtual machines
9	Containerization: a. Containerization technologies: docker, rocket, LXD b. Install and configure any containerization technology c. Create and use containers using it
10	Linux Kernel: Learn Linux kernel with respect to: a. What is Linux kernel? b. Operating model c. Licensing Model d. How does development work? e. Download kernel source code. f. Compile the Kernel

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** II	
Subject : Descriptive Statistics - I			
Period per Week(Each 60 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	2hrs	50
	Total	__	150

Course: UGCS205	Descriptive Statistics - I (Credits : 4 Lectures/Week: 4)	Lectures
	Expected Learning Outcomes: After successful completion of this course, students would be able to: 1. Identify the descriptive statistical concepts & present it graphically. 2. Recognize basic knowledge of R language. 3. Analyze the data and its properties by use of central tendency and variability. 4. Analyze the relationship between two quantitative variables using Correlation and Regression	
Unit I	**Data Presentation** **Data types :** attribute, variable, discrete and continuous variable, Univariate and Bivariate distribution. Types of Characteristics, Different types of scales: nominal, ordinal, interval and ratio. **Data presentation :** frequency distribution, histogram ogive, curves, stem and leaf display**.** **Introduction to R Programming**: introduction, Environment, Variable, Operators, Decision Making, Loops, Functions, String, Vectors, List, Matrices, Arrays, Factors, Data Frame, R Charts & Graphic, R Statistics	**15l**
Unit II	**Measures of Central tendency:** Mean, Median, mode for raw data, discrete, grouped frequency distribution. **Measures dispersion:** Variance, standard deviation, coefficient of variation for raw data, quartiles, quantiles Real life examples. **Moments:** raw moments, central moments, relation between raw and central moments. **Measures of Skewness and Kurtosis:** Concept of Skewness and Kurtosis, measures based on moments, quartiles.	**15l**
Unit III	**Correlation:** Concept of correlation, Types and interpretation, **Measure of Correlation:** Scatter diagram and interpretation; Karl Pearson's coefficient of correlation (r): Definition, examples for ungrouped and grouped data, effect of shift of origin and change of scale, properties; Spearman's rank correlation coefficient: Definition, examples of with and without repetition. Concept of Multiple correlation.	**15l**
Unit IV	**Regression:** Concept of dependent (response) and independent (predictor) variables, concept of regression, Types and prediction, difference between correlation and regression, Relation between correlation and regression. **Linear Regression** - Definition, examples using least square method and regression coefficient, coefficient of determination, properties. Concept of Multiple regression and Logistic regression.	**15l**
	Text Book: 1) Trivedi, K.S.(2001) : Probability, Statistics, Design of Experiments and Queuing Semester End Examination, with applications of Computer Science, Prentice Hall of India, New Delhi **Additional References:**	

	1) Ross, S.M. (2006): A First course in probability. 6th Edn Pearson 2) Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999): common statistical tests. Satyajeet Prakashan, Pune 3) Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics,S. Chand and Sons, New Delhi 4) Gupta, S.C. and Kapoor, V.K. (1999): Applied Statistics, S. Chand and Son's,New Delhi 5) Montgomery, D.C. (2001): Planning and Analysis of Experiments, wiley. **Links:** **1. https://www.tutorialspoint.com/statistics/hypothesis_testing.htm** **2. https://2012books.lardbucket.org/pdfs/beginning-statistics.pdf** **3. http://www.math.louisville.edu/~pksaho01/teaching/Math662TB-09S.pdf** **4.** https://pakistandasti.files.wordpress.com/2013/11/introduction-to-computer-theory-by-cohen-copy.pdf	

Sr.No	Practicals of UGCSP205 (To be implemented using R)
1	**Basics of R** a. Data input, Arithmetic Operators b. Vector Operations, Matrix Operations c. Data Frames, Built-in Functions
2	**Importing Data** a. R- CVS Files b. R-Excel Files c. R In Build Files
3	**Frequency distribution and data presentation** a. Frequency Distribution (Univariate data/ Bivariate data) b. Diagrams c. Graphs
4	**Measures of Central Tendency** a. Arithmetic Mean b. Median c. Mode d. Partition Values
5	**Measures dispersion** a. Range and Coefficient of range b. Quartile deviation and Coefficient of quartile deviation

	c. Standard deviation, Variance and Coefficient of variation (C.V.)
6	**Moments** a. Raw moments b. Central moments
7	**Measures of Skewness -** a. Karl Pearson's measure of Skewness b. Bowley's measure of Skewness c. Moment coefficient of Skewness
8	**Measures of Kurtosis** a. Moment coefficient of Kurtosis (Absolute measure) b. Moment coefficient of Kurtosis (Relative measure)
9	**Correlation** a. Karl Pearson's correlation coefficient b. Spearman's Rank correlation
10	**Regression** a. Method of least squares b. Using regression coefficients

Class: F.Y.BSC CS	**Branch**: Computer Science	**Semester:** II	
Subject : Physical Computing and IoT Programming			
Period per Week(Each 60 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	2hrs	50
	Total	—	150

Course: UGCS206	**Physical Computing and IoT Programming** **(Credits : 4 Lectures/Week: 4)**	**Lectures**
	Expected Course Outcomes: After successful completion of this course, students would be able to: 1. Interpret System On Chip Architectures 2. Prepare Raspberry Pi with hardware and installation.	

	3. Analyze physical interfaces and electronics of Raspberry Pi and program them using practical's 4. Examine how to make consumer grade IoT safe and secure with proper use of protocols	
Unit I	**SoC and Raspberry Pi** **System on Chip:** What is System on chip? Structure of System on Chip **SoC products:** FPGA**,** GPU, APU, Compute Units. **ARM 8 Architecture:** ARM 8 Architecture **Introduction to Raspberry Pi:** Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. **Raspberry Pi Boot:** Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.	**15l**
Unit II	**Programming Raspberry Pi** **Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands **Programing interfaces:** Introduction to Node.js, Python **Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.	**15l**
Unit III	**Introduction to IoT:** Sensing, Actuation, Networking basics, Sensor Networks, IoT Definition, Advantages, Disadvantages, IoT Functional Blocks, Physical design of IoT, Logical design of IoT, IOT Technology and Protocols, **IoT applications for industry**: Future Factory Concepts, Smart Objects, Smart Applications, IoT for Retailing Industry, IoT For Oil and Gas Industry, Home Management, eHealth, Simple IoT LED Program.	**15l**
Unit IV	**IoT and Protocols** Introduction, Privacy and Security Issues **IoT Security:** HTTP, UPnp, CoAP, MQTT, XMPP**.** **IoT Service as a Platform:** Clayster, Thinger.io, SenseIoT, carriots and Node RED. **IoT Security and Interoperability:** Risks, Modes of Attacks, Tools for Security and Interoperability.	**15l**
	Textbook(s): 1. Learning Internet of Things, Peter Waher, Packt Publishing(2015) 2. Mastering the Raspberry Pi, Warren Gay, Apress(2014) **Additional Reference(s):** 1. Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly **Links:** 1) http://ebook.pldworld.com/_eBook/-Packt%20Publishing%20Limited-/9781783553532-LEARNING_INTERNET_OF_THINGS.pdf 2) http://mensshed-llandudno.co.uk/wp-content/uploads/Mastering%20the%20Raspberry%20Pi.pdf	

Sr.No	Practicals of UGCSP206
1	Preparing Raspberry Pi: Hardware preparation and Installation
2	Linux Commands: Exploring the Raspbian
3	GPIO: Light the LED with Python
4	GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas
5	SPI: Camera Connection and capturing Images using SPI
6	Node RED: Connect LED to Internet of Things
7	Create a simple Web server using Raspberry Pi
8	Stepper Motor Control: PWM to manage stepper motor speed
9	Real Time Clock display using PWM
10	Configure & Test Audio on Raspberry Pi