



**RAMNIRANJAN JHUNJHUNWALA
COLLEGE OF ARTS, SCIENCE AND
COMMERCE, GHATKOPAR(W), MUMBAI
(AFFILIATED TO MUMBAI UNIVERSITY)**



**SYLLABUS FOR: F. Y. BSc
PROGRAM: B.Sc.
COURSE: COMPUTER SCIENCE**

WITH EFFECT FROM ACADEMIC YEAR 2019-20

Date: _____

Signature of BOS Members

1) Chairman :Anita Gaikwad

2) Subject Experts from outside the Parent University:

i) Prof Suchita Bhovar,
SNDT College, Ghatkopar

i) Prof Pratibha Deshmukh,
Bharathi Vidyapeeth, Navi Mumbai

3) Expert to be nominated by Vice Chancellor:

Prof Sampada Margaj, Kirti College Dadar

4) Representative from industry:

Mr Uday Pawar, Director Tech, People Interactive Pvt ltd

5) Post graduate alumni:

Prof Sunita Rai, Khalasa College, Matunga

6) Experts from outside the college:

- i) Prof Geeta Brijwani, K C College, Churchgate
- ii) Prof Maya Nair, SIES College, Sion
- iii) Prof Poonam Pandey, Somaiya College, Vidyavihar

Preamble

Information and Communication Technology (ICT) has today become integral part of all industry domains as well as fields of academics and research. The industry requirements and technologies have been steadily and rapidly advancing. Organizations are increasingly opting for open source systems. The students too these days are thinking beyond career in the industry and aiming for research opportunities.

The B.Sc. Computer Science course structure therefore needed a fresh outlook and complete overhaul. A real genuine attempt has been made while designing the new syllabus for this 3- year graduate course. Not only does it prepare the students for a career in Software industry, it also motivates them towards further studies and research opportunities.

The core philosophy of overall syllabus is to -

- a. Form strong foundation of Computer science,
- b. Introduce emerging trends to the students in gradual way,
- c. Groom the students for the challenges of ICT industry

In the first year i.e. for semester I & II, basic foundation of important skills required for software development is laid. The syllabus proposes to have four core subjects of Computer science and two core courses of Mathematics-Statistics. All core subjects are proposed to have theory as well as practical tracks. While the Computer Science courses will form fundamental skills for solving computational problems, the Mathematics & Statistics course will inculcate research- oriented acumen.

The syllabus design for further semesters encompasses more advanced and specialized courses of Computer Science.

We sincerely believe that any student taking this course will get very strong foundation and exposure to basics, advanced and emerging trends of the subject. We hope that the students' community and teachers' fraternity will appreciate the treatment given to the courses in the syllabus.

We wholeheartedly thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents; we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us.

Special thanks to Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly helped designing certain specialized courses and the syllabus as a whole.

Academic year 2018-2019
Semester – I

Paper Code	Course Type	Course Title	Credits	Lectures/Week
RJSCSS101	Core Subject	Computer Organization and Design	2	3
RJSCSS102	Core Subject	Programming with Python-I	2	3
RJSCSS103	Core Subject	Free and Open Source Software	2	3
RJSCSS104	Core Subject	Database Systems	2	3
RJSCSS105	Core Subject	Discrete Mathematics	2	3
RJSCSS106	Core Subject	Descriptive Statistics and Introduction to Probability	2	3
RJSCSS107	Ability Enhancement Course 1	Soft Skills Development	2	3
RJSCSS1P01	Core Subject Practical	Practical of RJSCSS101 + RJSCSS102	2	6
RJSCSS1P02	Core Subject Practical	Practical of RJSCSS103+RJSCSS104	2	6
RJSCSS1P03	Core Subject Practical	Practical of RJSCSS105+RJSCSS106	2	6

Semester – II

Paper Code	Course Type	Course Title	Credits	Lectures/Week
RJSCSS201	Core Subject	Programming with C	2	3
RJSCSS202	Core Subject	Programming with Python–II	2	3
RJSCSS203	Core Subject	Linux	2	3
RJSCSS204	Core Subject	Data Structures	2	3
RJSCSS205	Core Subject	Calculus	2	3
RJSCSS206	Core Subject	Statistical Methods and Testing of Hypothesis	2	3
RJSCSS207	Ability Enhancement Course 2	Green Technologies	2	3
RJSCSS2P01	Core Subject Practical	Practical of RJSCSS201 + RJSCSS202	2	6
RJSCSS2P02	Core Subject Practical	Practical of RJSCSS203+RJSCSS204	2	6
RJSCSS2P03	Core Subject Practical	Practical of RJSCSS205+RJSCSS206	2	6

Semester I - Theory

Course: RJSCSSS101	Computer Organization and Design (Credits : 2 Lectures/Week: 3)	
Objectives. To understand the structure and operation of modern processors and their instruction sets Expected Learning Outcomes: <ol style="list-style-type: none"> 1) To learn about how computer systems work and underlying principles 2) To understand the basics of digital electronics needed for computers 3) To understand the basics of instruction set architecture for reduced and complex instruction sets 4) To understand the basics of processor structure and operation 5) To understand how data is transferred between the processor and I/O devices 		
Unit I	Computer Abstractions and Technology: Basic structure and operation of a computer, functional units and their interaction. Representation of numbers and characters. Logic circuits and functions: Combinational circuits and functions: Basic logic gates and functions, truth tables; logic circuits and functions. Minimization with Karnaugh maps. Synthesis of logic functions with and-or-not gates, nand gates, nor gates. Fan-in and fan-out. Half adder, full adder. Sequential and combinational circuits and functions:	15 L
Unit II	FLIP-FLOP : SR , JK , RS, D flip flop. Shift registers Multiplexers and Demultiplexer. Instruction set architectures of 8086: Memory organization, addressing and operations; word size, big-endian and little-endian arrangements. Instruction sets for RISC and CISC. Operand addressing modes. Machine language, assembly language, assembler. instructions: arithmetic, logic, shift, etc. Instruction sets.	15 L
Unit III	Basic Processor Unit: Main components of a processor: registers and register files, ALU, control unit, instruction fetch unit, interfaces to instruction and data memories. Datapath. Instruction fetch and execute; executing arithmetic/logic, memory access and branch Basic I/O: Accessing I/O devices, data transfers between processor and I/O devices. Interrupts and exceptions: interrupt requests and processing.	15 L

Text book:

1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012

Additional References:

1. Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011
2. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010

Course: RJSCSSS102	Programming with Python- I (Credits : 2 Lectures/Week: 3)	
<p>Objectives The objective of this paper is to introduce various concepts of programming to the students using Python.</p> <p>Expected learning outcomes</p> <ol style="list-style-type: none"> 1) Students should be able to understand the concepts of programming before actually starting to write programs. 2) Students should be able to develop logic for Problem Solving. 3) Students should be made familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc. 4) Students should be able to apply the problem solving skills using syntactically simple 		
Unit I	<p>Introduction to python programming Reasons for Python as the learner's first programming language. Introduction to the IDLE interpreter (shell) and its documentation. Data:variables,Basic data types int, float, Boolean. Built-in function type. Assigning Values to Variables Python Basic syntax:Introduction to Interactive Mode Programming, Script Mode Programming, python Identifiers, Reserved Words in python, The role of indentation in python, Comments in Python. 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. I/O function: Input/output with print and input functions. The del(delete) statement Introduction to python Data Types ,Python String ,List ,Tuple, Dictionary elements within these types. Enumerate the function & methods of string, tuple, list, dictionaries.</p>	15 L

Unit II	<p>The conditional statements: if, if- else,if-elif-else.</p> <p>Python Loops:While loop,while-else,for-else,For Loop,Nested loop, The built-in function range()</p> <p>Loop Control Statements:The continue statement loop, the break statement</p> <p>Function in python:Advantages of functions, function parameters, formal parameters, actual parameters,global and local variables, Introduction to Anonymous functions</p>	15 L
Unit III	<p>List comprehensions</p> <p>Dictionaries: concept of key-value pairs, techniques to create, update and delete dictionary items.</p> <p>Gentle introduction to object-oriented programming with examples.</p> <p>Python Date-Time: Introduction to Time module functions.</p>	15 L
<p>Text books:</p> <ol style="list-style-type: none"> 1. Magnus Lie Hetland, Beginning Python: From Novice to Professional, Apress 2. Paul Gries, et al., Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Charles Dierbach, <i>Introduction to Computer Science using Python</i>, Wiley, 2013 2. Paul Gries , Jennifer Campbell, Jason Montojo, <i>Practical Programming: An Introduction to Computer Science Using Python 3</i>, Pragmatic Bookshelf, 2/E 2014 3. Adesh Pandey, <i>Programming Languages - Principles and Paradigms</i>, Narosa, 2008 		

Course: RJSCSSS103	Free and Open-source Software (Credits : 2 Lectures/Week: 3)	
<p>Objective: Open Source has acquired a prominent place in software industry. Having knowledge of Open Source and its related technologies is an essential for Computer Science student. This course introduces Open Source methodologies and ecosystem to students.</p> <p>Expected Learning Outcome:</p> <ol style="list-style-type: none"> 1) Upon completion of this course, students should have a good working knowledge of Open Source ecosystem, its use, impact and importance. 2) This course shall help student to learn Open Source methodologies, case studies with real life examples. 		
Unit I	<p>Introduction Introduction: Open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean no cost.</p> <p>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</p>	15L

	Social Impact Open source vs. closed source, , Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government.	
Unit II	Case Studies Example Projects: Apache web server, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, wordpress, GCC, GDB, github, Open Office. Study: Understanding the developmental models, licensings, mode of funding,commercial/non-commercial use. Open Source Hardware. Open source media. Collaboration, Community and Communication Contributing to Open Source Projects Introduction to github, interacting with the community on github, Communication and etiquette, testing open source code, reporting issues, contributing code. Introduction to wikipedia, contributing to Wikipedia Or contributing to any prominent open source project of student's choice. .	15L
Unit III	Understanding Open Source Ecosystem Open Source Operating Systems: GNU/Linux, Android, Open Solaris. Open Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, debuggers, Programming languages, LAMP, Open Source database technologies	15L
Text books: <ol style="list-style-type: none"> 1. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006 2. The official Ubuntu Book, 8th Edition Additional references: <ol style="list-style-type: none"> 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/ 		

Course: RJSCSS104	Database Systems (Credits : 2 Lectures/Week: 3)	
<p>Objectives: The objective of this course is to introduce the concept of the DBMS with respect to the relational model, to specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases</p> <p>Expected Learning Outcomes</p> <ol style="list-style-type: none"> 1) Students should be able to evaluate business information problem and find the requirements of a problem in terms of data. 2) Students should be able to design the database schema with the use of appropriate data types for storage of data in database. 3) Students should be able to create, manipulate, query and back up the databases. 		
Unit I	<p>Introduction to DBMS - Database, DBMS - Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture</p> <p>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)</p> <p>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</p> <p>ER to Table- Entity to Table, Relationship to tables with and without key constraints.</p>	15L
Unit II	<p>Schema refinement and Normal forms: Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition.</p> <p>Relational Algebra operations (selection, projection, set operations union, intersection, difference, cross product, Joins -conditional, equi join and natural joins, division)</p> <p>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, Backing Up and Restoring databases</p> <p>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</p>	15L

Textbook:

Unit III	<p>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)</p> <p>Joining Tables - inner join, outer join (left outer, right outer, full outer)</p> <p>Subqueries - subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries</p> <p>Database Protection: Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control</p> <p>Views (creating, altering dropping, renaming and manipulating views)</p> <p>DCL Statements (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges)</p> <p>TCL Statements Commit, Rollback and SavePoint</p>	15L
----------	---	-----

Text books:

1. Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Education, Sixth Edition, 2010
2. Ramakrishnam, Gehrke, Database Management Systems, McGraw-Hill, 2007
3. Joel Murach, Murach's MySQL, Murach, 2012

Additional References:

1. Robert Sheldon, Geoff Moes, Beginning MySQL, Wrox Press, 2005.

Course: RJSCSS105	Discrete Mathematics (Credits : 2 Lectures/Week: 3)	
------------------------------------	--	--

Objectives:

The purpose of the course is to familiarize the prospective learners with mathematical structures that are fundamentally discrete. This course introduces sets and functions, forming and solving recurrence relations and different counting principles. These concepts are useful to study or describe objects or problems in computer algorithms and programming languages.

Expected Learning Outcomes:

To provide overview of theory of discrete objects, starting with relations and partially ordered sets.

- 1) Study about recurrence relations, generating function and operations on them.
- 2) Give an understanding of graphs and trees, which are widely used in software.
- 3) Provide basic knowledge about models of automata theory and the corresponding formal languages.

Unit I	Recurrence Relations 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. Relations: Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Daigrams , Maximum and Minimum elements, Lattices Recurrence Relations: Definition of recurrence relations, Formulating recurrence relations, solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients. Solving linear homogeneous recurrence relations with constant coefficients of degree two when characteristic equation has distinct roots and only one root, Particular solutions of non linear homogeneous recurrence relation, Solution of recurrence relation by the method of generation functions, Applications – Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi	15L
Unit II	Counting Principles , Languages Permutations and Combinations: Partition and Distribution of objects, Permutation with distinct and indistinct objects, Binomial numbers, Combination with identities: Pascal Identity, Vandermonde's Identity, Pascal triangle, Binomial theorem, Combination with indistinct objects. Counting Principles: Sum and Product Rules, Two-way counting, Tree diagram for solving counting problems, Pigeonhole Principle (without proof); Simple examples, Inclusion Exclusion Principle (Sieve formula) (Without proof). Set Theory : Set definition, set builder notations, types of sets, Venn Diagrams, Set Operations, Power Set.	15L
Unit III	Graphs and Trees Graphs : Definition and elementary results, Adjacency matrix, path matrix, Representing relations using diagraphs, Warshall's algorithm- shortest path , Linked representation of a graph, Operations on graph with algorithms - searching in a graph; Insertion in a graph, Deleting from a graph, Traversing a graph- Breadth-First search and Depth-First search. Trees: Definition and elementary results. Ordered rooted tree, Binary trees, Complete and extended binary trees, representing binary trees in memory, traversing binary trees, binary search tree, Algorithms for searching and inserting in binary search trees, Algorithms for deleting in a binary search tree	15L

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.

Additional References:

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.

Course: RJSCS106	Descriptive Statistics and Introduction to Probability (Credits : 2 Lectures/Week: 3)	
Objectives: The purpose of this course is to familiarize students with basics of Statistics. This will be essential for prospective researchers and professionals to know these basics.		
Expected Learning Outcomes: <ol style="list-style-type: none"> 1) Enable learners to know descriptive statistical concepts 2) Enable study of probability concept required for Computer learners 		
Unit I	Data Presentation Data types : attribute, variable, discrete and continuous variable Data presentation : frequency distribution, histogram o give, curves, stem and leaf display Data Aggregation 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, discrete and grouped frequency distribution, quartiles, quantiles Real life examples	15L
Unit II	Moents:m raw moments, central moments, relation between raw and central moments Measures of Skewness and Kurtosis: based on moments, quartiles, relation between mean, median, mode for symmetric, asymmetric frequency curve. Correlation and Regression: bivariate data, scatter plot, correlation, nonsense correlation, Karl pearson's coefficients of correlation, independence. Linear regression: fitting of linear regression using least square regression, coefficient of determination, properties of regression coefficients (only statement)	15L
Unit III	Probability : Random experiment, sample space, events types and operations of events Probability definition : classical, axiomatic, Elementary Theorems of probability (without proof) <ul style="list-style-type: none"> – $0 \leq P(A) \leq 1$, – $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ – $P(A') = 1 - P(A)$ – $P(A) \leq P(B)$ if $A \subset B$ Conditional probability, 'Bayes' theorem, independence, Examples on Probability	15L

Probability:**Text Book:**

1. Trivedi, K.S.(2001) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi

- 1) Understand the importance and type of communication in personal and professional environment.
- 2) To provide insight into much needed technical and non-technical qualities in career planning.
- 3) Learn about Leadership, team building, decision making and stress management

Course:**RJSCSS107****Soft Skills Development
(Credits : 2 Lectures/Week: 3)****Objectives:**

To help learners develop their soft skills and develop their personality together with their technical skills. Developing professional, social and academic skills to harness hidden strengths, capabilities and knowledge equip them to excel in real work environment and corporate life. Understand various issues in personal and profession communication and learn to overcome them

Expected Learning Outcomes:

- 1) To know about various aspects of soft skills and learn ways to develop personality
- 2) Understand the importance and type of communication in personal and professional environment.
- 3) To provide insight into much needed technical and non-technical qualities in career planning
- 4) Learn about Leadership, team building, decision making and stress management

Unit I	<p>Introduction to Soft Skills and Hard Skills</p> <p>Personality Development: Knowing Yourself, Positive Thinking, Johari's Window, Communication Skills, Non-verbal Communication, Physical Fitness</p> <p>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</p> <p>Etiquette and Mannerism: Introduction, Professional Etiquette, Technology Etiquette</p> <p>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</p>	15L
Unit II	<p>Academic Skills</p> <p>Employment Communication: Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter</p> <p>Professional Presentation: Nature of Oral Presentation, Planning a Presentation, Preparing the Presentation, Delivering the Presentation</p> <p>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</p> <p>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</p>	15L
Unit III	<p>Professional Skills</p> <p>Creativity at Workplace: Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method</p> <p>Ethical Values: Ethics and Society, Theories of Ethics, Correlation between Values and Behavior, Nurturing Ethics, Importance of Work Ethics, Problems in the Absence of Work Ethics</p> <p>Capacity Building: Learn, Unlearn and Relearn: Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building</p> <p>Leadership and Team Building: Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams,</p> <p>Decision Making and Negotiation: Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts</p> <p>Stress and Time Management: Stress, Sources of Stress, Ways to Cope with Stress</p>	15L

Textbook:

1. *Soft Skills: an Integrated Approach to Maximise Personality*, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India

Additional References:

1. *Personality Development and Soft Skills*, Barun K. Mitra, Oxford Press
2. *Business Communication*, Shalini Kalia, Shailja Agrawal, Wiley India
3. *Soft Skills - Enhancing Employability*, M. S. Rao, I. K. International
4. *Cornerstone: Developing Soft Skills*, Sherfield, Pearson Indi

Semester I - Practical

Course: RJSCSS1P01	Practical of RJSCSS101 + RJSCSS102 (Credits : 6, Lectures/Week: 18)	
RJSCSS101	<p>Computer Organization and Design</p> <ol style="list-style-type: none"> 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 adder. 4. Design and verify a full adder. 5. Design and verify a half subtractor . 6. Design and verify half subtractor 7. Design a 4 bit magnitude comparator using combinational circuits. 8. Design and verify the operation of flip-flops using logic gates. 9. Verify the operation of a counter. 10. Verify the operation of a 4 bit shift register. 11. Assembly program for addition and subtraction. 12. Assembly program for multiplication and division. <p># Practical No. 1 to 8 can be performed using any open source simulator (like Logisim) (Download it from https://sourceforge.net/projects/circuit/)</p> <p># Practical No. 9 and 10 are required to be done using SPIM. SPIM is a self- contained simulator that will run MIPS R2000/R3000 assembly language programs.</p> <p># Latest version is available at https://sourceforge.net/projects/spimsimulator/</p>	

RJSCSS102	<p>Programming with Python – I</p> <ol style="list-style-type: none"> 1. Installing and setting up the Python IDLE interpreter. Executing simple statements like expression statement (numeric and Boolean types), assert, assignment, delete statements; the print function for output. 2. Script and interactive modes; defining a function in the two modes; executing a script; interactively executing a statement list (semicolon-separated sequence of simple statements); the input function. 3. Programs based on lists, conditional constructs, the for statement and the range function; interactively using the built-in functions len, sum, max, min 4. Programs related to string manipulation 5. Programs based on the while statement; importing and executing built-in functions from the time, math and random modules 6. Programs using break and continue statements. 7. Programs related to dictionaries 8. Programs using the built-in methods of the string, list and dictionary classes 	
RJSCSS103	<p>Free and Open Source Software</p> <ol style="list-style-type: none"> 1. Identify any Open Source software and create detailed report about it Sample Guidelines. <ol style="list-style-type: none"> a. Idea b. What problem does it solves? 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. <ol style="list-style-type: none"> a. History of license b. Idea c. What problems does it solve? d. Detailed licensing model e. Which popular software are released under this license? f. Any popular news associated with this license? g. Popularity h. Impact 	

	<p>3. Hands on with Open Source Software</p> <ul style="list-style-type: none"> 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 <p>4. Contributing to Wikipedia:</p> <ul style="list-style-type: none"> 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 <p>5. Github</p> <ul style="list-style-type: none"> a. Create and publish your own open source project: Write any simple program using your choice of 	
--	---	--

	<ul style="list-style-type: none"> b. Create a repository on github and save versions of your project. You'll learn about the staging area, committing your c. code, branching, and merging, d. 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. <p>6.Open Source Operating Systems</p> <ul style="list-style-type: none"> 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. <p>7. Virtualization: Open Source virtualization technologies:</p> <ul style="list-style-type: none"> a. Install and configure any one: VirtualBox. b. Create and use virtual machines <p>8. Containerization:</p> <ul style="list-style-type: none"> a. Containerization technology: docker <p>9. Linux Kernel: Learn Linux kernel with respect to:</p> <ul style="list-style-type: none"> a. How development works? b. Download kernel source code. c. Compile the Kernel 	
RJSCSS1 04	<p>Database Systems</p> <ol style="list-style-type: none"> 1. For given scenario <ul style="list-style-type: none"> • Draw E-R diagram and convert entities and relationships to table. 2. Write relational algebra queries on the tables created in Practical-1. 3. Perform the following: <ul style="list-style-type: none"> • 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 4. Perform the following: <ul style="list-style-type: none"> • Altering a Table • Dropping/Truncating/Renaming Tables • Backing up / Restoring a Database 	

	<p>Perform the following:</p> <ul style="list-style-type: none"> • Simple Queries • Simple Queries with Aggregate functions • Queries with Aggregate functions (group by and having clause) <p>Queries involving</p> <ul style="list-style-type: none"> • Date Functions • String Functions • Math Functions <p>Join Queries</p> <ul style="list-style-type: none"> • Inner Join • Outer Join <p>Subqueries</p> <ul style="list-style-type: none"> • With IN clause • With EXISTS clause <p>Views</p> <ul style="list-style-type: none"> • Creating Views (with and without check option) • Dropping views • Selecting from a view <p>TCL statements</p> <ul style="list-style-type: none"> • Saving (Commit) and Undoing (rollback), SavePoint <p>DCL statements</p> <ul style="list-style-type: none"> • Granting and revoking permissions 	
Course: RJSCSS1P0 3	Practical of RJSCSS105+ RJSCSS106 (Credits : 6, Lectures/Week: 18)	
RJSCSS105	<p>Discrete Mathematics</p> <p>Graphs of standard functions such as absolute value function, inverse function, logarithmic and exponential functions, flooring and ceiling functions, trigonometric functions over suitable intervals.</p> <p>Partial ordering sets, Hasse diagram and Lattices.</p> <p>Recurrence relation.</p> <p>Different counting principles.</p> <p>Finite state Automata and Finite state machines.</p> <p>Warshall's Algorithm.</p> <p>Shortest Path algorithms.</p> <p>Operations on graph.</p> <p>Breadth and Depth First search algorithms.</p> <p>Concept of searching, inserting and deleting from binary search trees.</p>	

RJSCSS106	Descriptive Statistics and Introduction to Probability (<i>To be implemented using R</i>)		
	<p>Frequency distribution and data presentation</p> <p>Measures of central tendency</p> <p>Data entry using, functions, c(), scan (), Creating vectors, Mathematical</p> <p>Operations: ** +/~/ / ^ , exp, log, log10, etc, creating vector of text type, useful functions: data, frame, matrix operations, seq(), split() etc.</p> <p>Frequency distribution using cut(), table()</p> <p>Data presentation</p> <p>Summary Statistics (measures of central tendency, dispersion)</p> <p>Measures of skewness and kurtosis</p> <p>Correlation and regression</p> <p>Probability</p> <p>Conditional probability</p>		

Semester II - Theory

Course: RJCSSS201	Programming with C (Credits : 2 Lectures/Week: 3)	
-----------------------------	--	--

Objectives:

The objective of this course is to provide a comprehensive study of the C programming language, stressing upon the strengths of C, which provide the students with the means of writing modular, efficient, maintainable, and portable code.

Expected Learning Outcomes

- 1) Students should be able to write, compile and debug programs in C language.
- 2) Students should be able to use different data types in a computer program.
- 3) Students should be able to design programs involving decision structures, loops and functions.
- 4) Students should be able to explain the difference between call by value and call by reference
- 5) Students should be able to understand the dynamics of memory by the use of pointers.
- 6) Students should be able to use different data structures and create/update basic data files.**

Unit-I	<p>Structure of C program: Header and body, Use of comments. Interpreters vs compilers, Python vs C. Compilation of a program. Formatted I/O: printf(), scanf().</p> <p>Data: Variables, Constants, data types like: int, float char, double and void, short and long size qualifiers, signed and unsigned qualifiers. Compare with datatypes in Python.</p> <p>Compare static typing in C vs dynamic typing in Python</p> <p>Variables: Declaring variables, scope of the variables according to block, hierarchy of data types. Compare explicit declarations in C with implicit declarations in Python.</p> <p>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.</p> <p>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.</p>	
--------	--	--

Unit-II	<p>Arrays: (One and two dimensional), declaring array variables, initialization of arrays, accessing array elements. Compare array types of C with list and tuple types of Python.</p> <p>Data Input and Output functions: Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts().</p> <p>Manipulating Strings: Declaring and initializing String variables, Character and string handling functions. Compare with Python strings.</p> <p>Functions: Function declaration, function definition, Global and local variables, return statement, Calling a function by passing values.</p> <p>Recursion: Definition, Recursive functions.</p>	
Unit-III	<p>Pointer: Fundamentals, Pointer variables, Referencing and de-referencing, Pointer Arithmetic, Using Pointers with Arrays, Using Pointers with Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers.</p> <p>Dynamic Memory Allocation: malloc(), calloc(), realloc(), free() and sizeof operator. Compare with automatic garbage collection in Python.</p> <p>Structure: Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures. Compare C structures with Python tuples</p> <p>Unions: Defining and working with unions.</p> <p>File handling: Different types of files like text and binary, Different types of functions: fopen(), fclose(), fgetc(), fputc(), fgets(), fputs(), fscanf(), fprintf(), getw(), putw(), fread(), fwrite(), fseek().</p> <p>Command Line Arguments</p>	

Text books:

1. Programming in ANSI C (Third Edition) : E Balagurusamy, TMH

Additional References:

1. Pradip Dey, Manas Ghosh, "Programming in C", second edition, Oxford University Press
2. Yashavant P. Kanetkar. "Let Us C", BPB Publications

Course: RJCSSS202	Programming with Python - II (Credits : 2 Lectures/Week: 3)	
Objective: The objective of this paper is to explore the style of structured programming to give the idea to the students how programming can be used for designing real-life applications by reading/writing to files, GUI programming, interfacing database/networks and various other features.		
Expected Learning Outcomes <ol style="list-style-type: none"> 1) Students should be able to understand how to read/write to files using python. 2) Students should be able to catch their own errors that happen during execution of programs. 3) Students should get an introduction to the concept of pattern matching. 4) Students should be made familiar with the concepts of GUI controls and designing GUI applications. 5) Students should be able to connect to the database to move the data to/from the application. 6) 6)Students should know how to connect to computers, read from URL and send email. 		
Unit I	Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problemsolving applications. Exception handling: What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise. Regular Expressions: various types of regular expressions & functions.	15 L
Unit II	GUI Programming in Python (using Tkinter/wxPython/Qt) What is GUI, Advantages of GUI, Introduction to GUI library. Layout management, events and bindings, fonts, colours, drawing on canvas (line, oval, rectangle, etc.) Widgets such as : frame, label, button, checkbutton, entry, listbox, message, radiobutton, text, spinbox etc	15 L
Unit III	Database connectivity in Python: Installing mysql connector, accessing connector module module, using connect, cursor, execute & close functions, reading single & multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in database connectivity. Network connectivity: Socket module, creating server-client programs, sending email, reading from URL	15 L
Text books: <ol style="list-style-type: none"> 1. Paul Gries , Jennifer Campbell, Jason Montojo, <i>Practical Programming: An Introduction to Computer Science Using Python 3</i>, Pragmatic Bookshelf, 2/E 2014 Additional References: <ol style="list-style-type: none"> 1. James Payne , <i>Beginning Python: Using Python 2.6 and Python 3</i>, Wiley India, 2010 2. A. Lukaszewski, <i>MySQL for Python: Database Access Made Easy</i>, Pact Publisher, 2010 		

Course: RJCSS203	Linux (Credits : 2 Lectures/Week: 3)	
Objectives: This course introduces various tools and techniques commonly used by Linux programmers, system administrators and end users to achieve their day to day work in Linux environment. It is designed for computer students who have limited or no previous exposure to Linux.		
Expected Learning Outcomes: <ol style="list-style-type: none"> 1) Upon completion of this course, students should have a good working knowledge of Linux, from both a graphical and command line perspective, allowing them to easily use any Linux distribution. 2) This course shall help student to learn advanced subjects in computer science practically. 3) Student shall be able to progress as a Developer or Linux System Administrator using the acquired 		
Unit I	Linux System: History, Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management, File system, I/O. Linux Basics: Looking into the Linux Kernel, GNU Utilities, Desktop environments, The Linux console The Unix/Linux architecture, Features of Unix/Linux. Basic bash shell commands: Starting the shell, Shell prompt, File system Navigation, File and directory listing, File handling, Directory handling, Viewing file contents More bash shell commands: Monitoring programs, Monitoring disk space, Working with data files: Sorting, Searching, Compressing, Archiving The Linux environment variable: Environment variables, setting environment variables, Removing environment variables, Default shell environment variables, setting the PATH environment variables, Using command aliases	15L
Unit II	Understanding Linux file permission: Linux security, Using Linux groups, Decoding file permissions, Changing security settings vi editor Basic script building: Using multiple commands, Creating a script file, Displaying messages, Using variables, Redirecting Input and Output, Pipes, Performing math, Exiting the script. Using structured commands: Working with the if-then, if-then-else and nesting if statements, test command, Compound condition testing, advanced if then features, the case command. More structured commands: for command, C-style for command, while command, until command, nesting loops, Looping on file data, controlling the loop, processing the o/p of a loop.	15L
Unit III	Script control: Handling signals, Running scripts in background mode, Job control, Job Scheduling Commands: nice, renice, at, batch, cron table. Editors: Sed and awk Essential System Administration: root: The system administrator's login, run levels, The administrator's privileges, Startup & Shutdown. TCP/IP networking: TCP/IP Basics, TCP/IP Model, Resolving IP addresses, Applications, telnet, ftp, Berkeley commands.	15L

Text book:

- 1) Unix Concepts and Applications by Sumitabha Das.
- 2) Official Ubuntu Book, 8th Edition, by Matthew Helmke & Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew, Prentice Hall

Additional References:

- 1) Linux kernel Home: <http://kernel.org>
- 2) Open Source Initiative: <https://opensource.org/>
- 3) The Linux Foundation: <http://www.linuxfoundation.org/>

Course RJSCSS204	Data Structures (Credits : 2 Lectures/Week: 3)	
Objectives: To explore and understand the concepts of Data Structures and its significance in programming. Provide and holistic approach to design, use and implement abstract data types. Understand the commonly used data structures and various forms of its implementation for different applications using Python. Expected Learning Outcomes: <ol style="list-style-type: none"> 1) Learn about Data structures, its types and significance in computing 2) Explore about Abstract Data types and its implementation 3) Ability to program various applications using different data structure in Python 		
Unit I	Abstract Data Types: Introduction, The Date Abstract Data Type, Bags, Iterators. Application Arrays: Array Structure, Python List, Two Dimensional Arrays, Multi Dimensional Arrays. Sets and Maps: Sets-Set ADT, Selecting Data Structure, List based Implementation, Maps-Map ADT, List Based Implementation Searching and Sorting: Searching-Linear Search, Binary Search, Sorting-Bubble, Selection Working with Sorted Lists-Maintaining Sorted List, Maintaining sorted Lists.	15L
Unit II	Linked Structures: Introduction, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, Linked List Implementation. Stacks: Stack ADT, Implementing Stacks-Using Python List, Using Linked List, Stack Applications-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-III	<p>Advanced Linked List: Doubly Linked Lists-Organization and Operation, Circular Linked List-Organization and Operation.</p> <p>Recursion: Recursive Functions, Properties of Recursion, Its working, Recursive Applications</p> <p>Hash Table: Introduction, Hashing-Linear Probing, Clustering, Rehashing, Separate Chaining, Hash Functions.</p>	15L
----------	---	------------

Text book:

- 1) *Data Structure and algorithm Using Python*, Rance D. Necaie, 2016 Wiley India Edition
- 2) *Data Structure and Algorithm in Python*, Michael T. Goodrich, Robertom Tamassia, M. H. Goldwasser, 2016 Wiley India Edition

Additional 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

Course: RJSCSS205	Calculus (Credits : 2 Lectures/Week: 3)	
Objectives: The course is designed to have a grasp of important concepts of Calculus in a scientific way. It covers topics from as basic as definition of functions to partial derivatives of functions in a gradual and logical way. The learner is expected to solve as many examples as possible to get complete clarity and understanding of the topics covered.		
Expected Learning Outcomes: 1) Understanding of Mathematical concepts like limit, continuity, derivative, integration of functions.		
Unit I	DERIVATIVES AND ITS APPLICATIONS: Review of Functions, limit of a function, continuity of a function, derivative function. Derivative In Graphing And Applications: Analysis of Functions: Increase, Decrease, Concavity, Relative Extrema; Graphing Polynomials, Rational Functions, Cusps and Vertical Tangents. Absolute Maxima and Minima, Applied Maximum and Minimum Problems,	15L
Unit II	INTEGRATION AND ITS APPLICATIONS: An Overview of the Area Problem, Indefinite Integral, Definition of Area as a Limit; Sigma Notation, Definite Integral, Evaluating Definite Integrals by Substitution, Area Between Two Curves, Length of a Plane Curve. Numerical Integration: Simpson's Rule. Modeling with Differential Equations, Separation of Variables, Slope Fields, Euler's Method, First- Order Differential Equations	15L
Unit III	PARTIAL DERIVATIVES AND ITS APPLICATIONS: Functions of Two or More Variables Limits and Continuity Partial Derivatives, Differentiability, Differentials, and Local Linearity, Chain Rule, Directional Derivatives and Gradients, Tangent Planes and Normal, Vectors, Maxima and Minima of Functions of Two	15L

Textbook:

1. *Calculus: Early transcendental (10th Edition)*: Howard Anton, Irl Bivens, Stephen Davis, John Wiley & sons, 2012.

Additional References:

1. *Calculus and analytic geometry (9th edition)*: George B Thomas, Ross L Finney, Addison Wesley, 1995
2. *Calculus: Early Transcendentals (8th Edition)*: James Stewart, Brooks Cole, 2015.

3. Calculus (10th Edition): Ron Larson, Bruce H. Edwards, Cengage Learning, 2013.
4. Thomas' Calculus (13th Edition): George B. Thomas, Maurice D. Weir, Joel R. Hass, Pearson, 2014.

Course: RJSCSS206	Statistical Methods and Testing of Hypothesis (Credits : 2 Lectures/Week: 3)	
Objectives: The purpose of this course is to familiarize students with basics of Statistics. This will be essential for prospective researchers and professionals to know these basics.		
Expected Learning Outcomes: <ol style="list-style-type: none"> 1) Enable learners to know descriptive statistical concepts 2) Enable study of probability concept required for Computer learners 		
Unit I	Standard distributions: random variable; discrete, continuous, expectation and variance of a random variable, pmf, pdf, cdf, reliability, Introduction and properties without proof for following distributions; binomial, normal, chi-square, t, F. Examples	15L
Unit II	Hypothesis testing: one sided, two sided hypothesis, critical region, p-value, tests based on t, Normal and F, confidence intervals.	15L
Unit III	Non-parametric tests: need of non-parametric tests, sign test, Wilcoxon's signed rank test, run test, Kruskal-Wallis tests. Post-hoc analysis of one-way analysis of variance : Duncan's test Chi-square test of	15L
Text Book: <ol style="list-style-type: none"> 1. Trivedi, K.S.(2009) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi Additional References: <ol style="list-style-type: none"> 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. (2002) : Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi 4. Gupta, S.C. and Kapoor, V.K. (4th Edition) : Applied Statistics, S. Chand and Son's, New Delhi 		

Course: RJSCSS207	Green Technologies (Credits : 2 Lectures/Week: 3)	
Objectives: To familiarize with the concept of Green Computing and Green IT infrastructure for making computing and information system environment sustainable. Encouraging optimized software and hardware designs for development of Green IT Storage, Communication and Services. To highlight useful approaches to green IT initiatives.		
Expected Learning Outcomes: 1) Learn about green IT can be achieved in and by hardware, software, network communication and data center operations. 2) Understand the strategies, frameworks, processes and management of green IT		
Unit I	Green IT Overview: Introduction , Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Green I , Holistic Approach to Greening IT, Greening IT, Applying IT for Enhancing Environmental Sustainability, Green IT Standards and Eco-Labeling of IT , Enterprise Green IT Strategy, Green Washing, Green IT: Burden or Opportunity? Green Devices and Hardware: Introduction , Life Cycle of a Device or Hardware, Reuse, Recycle and Dispose Green Software: Introduction , Processor Power States , Energy-Saving Software Techniques, Evaluating and Measuring Software Impact to Platform Power Sustainable Software Development: Introduction, Current Practices, Sustainable Software, Software Sustainability Attributes, Software	15L
Unit II	Green Data Centres: Data Centres and Associated Energy Challenges, Data Centre IT Infrastructure, Data Centre Facility Infrastructure: Implications for Energy Efficiency, IT Infrastructure Management, Green Data Centre Metrics Green Data Storage: Introduction , Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management Green Networks and Communications: Introduction, Objectives of Green Network Protocols, Green Network Protocols and Standards Enterprise Green IT Strategy: Introduction, Approaching Green IT Strategies, Business Drivers of Green IT Strategy, Business Dimensions for Green IT Transformation, Organizational Considerations in a Green IT	15L
Unit III	Sustainable Information Systems and Green Metrics: Introduction, Multilevel Sustainable Information, Sustainability Hierarchy Models, Product Level Information, Individual Level Information, Functional Level Information, Organizational Level Information, Measuring the Maturity of Sustainable ICT Enterprise Green IT Readiness: Introduction, Readiness and Capability, Development of the G-Readiness Framework, Measuring an Organization's G-Readiness Sustainable IT Services: Creating a Framework for Service Innovation: Introduction, Factors Driving the Development of Sustainable IT, Sustainable IT Services (SITS), SITS Strategic Framework Enterprise Greening, Information Systems in Greening Enterprises, Enterprise. IT Usage and Hardware, Inter-organizational Enterprise Activities and Green Issues	15L

Text book:

- 1) *Harnessing Green IT: Principles and Practices*, San Murugesan, G. R. Ganadharan, Wiley & IEEE.

Additional References:

- 2) *Green IT*, Deepak Shikarpur, Vishwkarma Publications, 2014
- 3) *Green Communications: Principles, Concepts and Practice*- Samdanis et al, J. Wiley
- 4) *Green IT for Sustainable Business Practice: An ISEB Foundation Guide*, Mark G. O'Neill, The Chartered Institute for IT, 2010

Course: RJSCSS2P01	Practical of RJSCSS201 + RJSCSS202	
RJSCSS201	Programming with C <ol style="list-style-type: none"> 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,DMA. 9. Programs on string manipulations. 10. Programs on basic file operations. 11. Programs on Command Line Arguments. 	
RJSCSS202	Programming with Python-II <ol style="list-style-type: none"> 1. Programs to read and write files. 2. Programs with iterables and iterators. 3. Program to demonstrate exception handling. 4. Program to demonstrate the use of regular expressions. 5. Program to show draw shapes 6. Program to demonstrate GUI controls. 7. Program to demonstrate database connectivity. 8. Program to create server-client and exchange basic information. 	

Course: RJSCSS2P02	Practical of RJSCSS203 + RJSCSS204	
RJSCSS203	<p>Linux</p> <ol style="list-style-type: none"> 1. Linux installation 2. Basic Linux commands such as file and directory manipulation, redirection and piping 3. Basic filter commands such as head, tail, more, cat, sort, cut, grep 4. Advanced filters such as egrep, fgrep, tr, sed, awk 5. File operation commands such as – split, tar, find, zip, ln, chmod 6. Basic shell scripting such as – defining variables, reading user input, conditions, loops, string operations, arithmetic operations 7. Advanced shell scripting such as - environment variables, shell features, command line arguments, file tests, using backticks, 8. Process management such as –ps, jobs, nice, fg, bg, at 9. Linux system administration such as – user management, mounting, job control(crontab), chown, chgrp etc 	
RJSCSS204	<p>Data structures</p> <ol style="list-style-type: none"> 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 <ol style="list-style-type: none"> a. Bubble sort b. selection 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 Postfix Evaluation. 7) Implement of a queue as a list which you add and delete items from. 8) Implement Linked list and demonstrate the functionality to add and delete items in the linked list. 9)Write a program for simple tree implementation. 10) Recursive implementation of <ol style="list-style-type: none"> a. Factorial b. Fibonacci 	

Course: RJSCSS2P03	Practical of RJSCSS205 + RJSCSS206	
RJSCSS205	Calculus <ol style="list-style-type: none"> 1. Continuity of functions; Derivative of functions 2. Increasing, decreasing, concave up and concave down functions 3. Relative maxima, relative minima, absolute maxima, absolute minima 4. Newton's method to find approximate solution of an equation 5. Area as a limit and length of a plane curve 6. Numerical integration using Simpson's rule 7. Solution of a first order first degree differential equation, Euler's method 8. Calculation of Partial derivatives of functions 9. Local linear approximation and directional derivatives 10. Maxima and minima of functions of two variables 	
RJSCSS206	Statistical Methods and Testing of Hypothesis <ol style="list-style-type: none"> 1. Problems based on binomial distribution 2. Problems based on normal distribution 3. Property plotting of binomial distribution 4. Property plotting of normal distribution 5. Plotting pdf, cdf, pmf, for discrete and continuous distribution 6. t test, normal test, F test 7. Analysis of Variance 8. Non parametric tests- I 9. Non- Parametric tests - II 10. Post-hoc analysis of one-way analysis 	

valuation Scheme

Internal Exam-20 Marks

(i) Internal-I - 20 Marks (objective base)

20 marks Test - Duration 30 mins

I. Internal-II Exam-20 Marks (subjective Base)

II. External Examination- 60 Marks

(i) Duration - 2.5 Hours.

(ii) Theory question paper pattern:-

All questions are compulsory.		
Question	Based on	Marks
Q.1	Unit I	15
Q.2	Unit II	15
Q.3	Unit III	15
Q.4	Unit I,II and III	15

III. Practical Examination - (50 marks x 6 core papers)

- Each core subject carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75 % practical from each core subjects are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Exam)