Resolution No: AC/II(21-22).2.RUS6

S. P. Mandali's

Ramnarain Ruia Autonomous College

(Affiliated to University of Mumbai)

Syllabus for FYB.Sc.

Program: B.Sc (Computer Science)

Program Code: Computer Science (RUSCS)

(Credit Based Semester and Grading System
for academic year 2021–2022)

PROGRAM OUTCOMES

PO	PO Description **A student completing Bachelor's Degree in Computer Science program will be able to:**
PO 1	Recall and explain acquired scientific knowledge in a comprehensive manner and apply the skills acquired in their chosen discipline. Interpret scientific ideas and relate its interconnectedness to various fields in science.
PO 2	Evaluate scientific ideas critically, analyse problems, explore options for practical demonstrations, illustrate work plans and execute them, organise data and draw inferences.
PO 3	Explore and evaluate digital information and use it for knowledge upgradation. Apply relevant information so gathered for analysis and communication using appropriate digital tools.
PO 4	Ask relevant questions, understand scientific relevance, hypothesize a scientific problem, construct and execute a project plan and analyse results.
PO 5	Take complex challenges; work responsibly and independently, as well as in cohesion with a team for completion of a task. Communicate effectively, convincingly and in an articulate manner.
PO 6	Apply scientific information with sensitivity to values of different cultural groups. Disseminate scientific knowledge effectively for upliftment of the society.
PO 7	Follow ethical practices at workplace and be unbiased and critical in interpretation of scientific data. Understand the environmental issues and explore sustainable solutions for it.
PO 8	Keep abreast with current scientific developments in the specific discipline and adapt to technological advancements for better application of scientific knowledge as a lifelong learner

PROGRAM SPECIFIC OUTCOMES

PSO	Description **A student completing Bachelor's Degree in Science program in the subject of Computer Science will be able to:**
PSO 1	Apply knowledge of computational mathematics ,statistics and programming acquired in the field of Computer Science.
PSO 2	Identify , analyze complex problems in the real world and formulate innovative solutions to those problems.
PSO 3	Compare and apply hardware and software technologies for implementing reliable optimized solutions catering to need and available resources.
PSO 4	Apply domain expertise to pursue higher education and Research in computer science discipline.
PSO 5	Apply software development, managerial, Professional and soft skills in industry
PSO 6	Understand the global needs and prepare themselves for the changing needs worldwide adapting an ability to engage in life- long learning..
PSO 7	Become a responsible ,ethical citizen and explore environmental issues to develop sustainable solutions for it.
PSO 8	Use the techniques, skills and modern computing tools to emerge as a freelancer and entrepreneur in the field.

PROGRAM OUTLINE

F.Y.B.Sc (Computer Science)							
SEMESTER – I (THEORY)					SEMESTER – I (PRACTICALS)		
YEAR	**SEM**	**COURSE CODE**	**COURSE TITLE**	**CREDITS**	**COURSE CODE**	**COURSE TITLE**	**CREDITS**
F.Y.Bsc	**I**	**RUSCS101**	Fundamentals of Computer Organization & Introduction to Embedded Systems	2	**RUSCSP101**	Practical of Fundamentals of Computer Organization& Introduction to Embedded Systems	1
F.Y.Bsc	**I**	**RUSCS102**	Programming with Python- I	2	**RUSCSP102**	Practical of Programming with Python- I	1
F.Y.Bsc	**I**	**RUSCS103**	Linux Fundamentals	2	**RUSCSP103**	Practical of Linux Fundamentals	1
F.Y.Bsc	**I**	**RUSCS104**	Algorithms and Programming with C	2	**RUSCSP104**	Practical of Algorithms and Programming with C	1
F.Y.Bsc	**I**	**RUSCS105**	Discrete Mathematics	2	**RUSCSP105**	Practical of Discrete Mathematics	1
F.Y.Bsc	**I**	**RUSCS106**	Descriptive Statistics and Introduction to probability	2	**RUSCSP106**	Practical of Descriptive Statistics and Introduction to probability	1
F.Y.Bsc	**I**	**RUSCS107**	Soft Skills Development	2	---	----	----

F.Y.B.Sc (Computer Science)							
SEMESTER – II (THEORY)					**SEMESTER – II (PRACTICALS)**		
YEAR	**SEM**	**COURSE CODE**	**COURSE TITLE**	**CREDITS**	**COURSE CODE**	**COURSE TITLE**	**CREDITS**
F.Y.Bsc	**II**	**RUSCS201**	Database Management Systems	2	**RUSCSP201**	Practical of Database Management Systems	1
F.Y.Bsc	**II**	**RUSCS202**	Programming with Python- II	2	**RUSCSP202**	Practical of Programming with Python- II	1
F.Y.Bsc	**II**	**RUSCS203**	Linux Server Administration	2	**RUSCSP203**	Practical of Linux Server Administration	1
F.Y.Bsc	**II**	**RUSCS204**	Data Structures	2	**RUSCSP204**	Practical of Data Structures	1
F.Y.Bsc	**II**	**RUSCS205**	Calculus	2	**RUSCSP205**	Practical of Calculus	1
F.Y.Bsc	**II**	**RUSCS206**	Statistical Methods	2	**RUSCSP206**	Practical of Statistical Methods	1
F.Y.Bsc	**I**	**RUSCS207**	Green Technologies	2	-----	-------	-------

Course Code: RUSCS101

Course Title: FUNDAMENTALS OF COMPUTER ORGANIZATION & INTRODUCTION TO EMBEDDED SYSTEMS

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Explains and demonstrate the architecture, structure, working, issues and problems, CPU architecture and its working.
CO 2	Demonstration to digital circuits.
CO 3	Illustrate the multicore systems and embedded systems along with its applications.
CO 4	Understand and explain the underlying principles of computers.
CO 5	Identify various hardware used in the computer.
CO 6	Construct how digital circuits are implemented in the computers.
CO 7	Simulates how data is transferred between various peripheral devices in the computer.

DETAILED SYLLABUS

Course Code	Unit	FUNDAMENTALS OF COMPUTER ORGANIZATION & INTRODUCTION TO EMBEDDED SYSTEMS	Lectures
RUSCS101	**I**	**Basic Structure of Computers -** Computer Types, Functional Units, Basic Operational Concepts, Performance, Historical Perspective, Design for Performance **A top Level View of Computer Function & Interconnection -** Computer components, Computer Function, Interconnection Structures, Bus Interconnection, PCI **The Memory System -** Basic concepts, Semiconductor RAM Memories, ROMs, DMA, Memory Hierarchy, Cache Memory, Performance Considerations, Virtual Memory, Memory Management Requirements, Secondary Storage **Instruction Set Architecture -** Memory locations and addresses, Memory operations, Instructions and Instruction sequencing, addressing modes, assembly language, stacks, subroutines, additional instructions, CISC instruction sets, RISC & CISC styles **Basic Processing Unit -**	15 L

		Fundamental concepts, Instruction Execution, Hardware components, Instruction Fetch & Execution steps, control signals, hardwired control, CISC-Style **Input/output Organizations -** Accessing I/O devices, Interrupts, Bus Structure, Bus operations, arbitration, interface circuits, interconnection standards	
	II	**Digital Circuits - Fundamental Concepts -** Introduction, Digital signals, basic digital circuits, NAND and NOR operations, Exclusive-OR and Exclusive-NOR operations, Boolean Algebra, Examples of IC Gates. **Number Systems & Codes -** Introduction, Number Systems, Binary Number System, Signed Binary Numbers, Binary Arithmetic, 2's Complement Arithmetic, Octal Number System, Hexadecimal Number System, codes **Combinational Logic Design -** Introduction, Standard representation for logic functions, Karnaugh Map Representation of Logic Functions, Simplification of logic functions using K-Map, minimization of logic function specified in minterm/maxterm or truth table, minimization of logic functions not specified in minterms/maxterms, Don't care conditions **Combination Logic Design Using MSI Circuits -** Introduction, Multiplexers-Demultiplexers-Decoders and their use, Adders and their use, BCD Arithmetic. **Flip-Flops -** Introduction, A-1 Bit memory cell, Clocked S-R Flip Flop, J-K Flip-Flop, D-type Flip-Flop, T-Type Flip-Flop.	15 L
	III	**Processing And Performance -** Hardware Multithreading, vector (SIMD) processing, Shared-Memory Multiprocessors, Cache Coherence, Message-Passing Multicomputer, Parallel Programming for Multiprocessors, Performance Modeling. **Multicore Computers -** Hardware performance issues, software performance issues, Multicore organization, Intel x86 Multicore Organization. **Introduction to Embedded Systems -**Introducing Embedded Systems, Philosophy, Embedded Systems, Embedded Design and Development Process.	15 L

Course Code	PRACTICAL OF FUNDAMENTALS OF COMPUTER ORGANIZATION & INTRODUCTION TO EMBEDDED SYSTEMS	Credits
RUSCSP101	1. Knowledge of hardware that goes in the making of a computer: Assembling of PC. Installation of OS, setting up of dual boot, installation of hardware and software. 2. Execution of File handling commands in DOS Prompt. 3. Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR). 4. Design and verify a half/full adder 5. Design and verify half/full subtractor 6. Design and verify the operation of flip-flops using logic gates. 7. Verify the operation of a counter.	**1**

	8. Verify the operation of a 4 bit shift register 9. Using SPIM, write and test an adding machine program that repeatedly reads in integers and adds them into a running sum. The program should stop when it gets an input that is 0, printing out the sum at that point. 10. Using SPIM, write and test a program that reads in a positive integer using the SPIM system calls. If the integer is not positive, the program should terminate with the message “Invalid Entry”; otherwise the program should print out the names of the digits of the integers, delimited by exactly one space. For example, if the user entered “528,” the output would be “Five Two Eight” # Practical No. 3 to 8 can be performed using any open source simulator (like Logisim) (Download it from https://sourceforge.net/projects/circuit/) # Practical No. 9 and 10 are required to be done using SPIM. #Latest version is available at https://sourceforge.net/projects/spimsimulator/	

References:

1. Computer Organization & Architecture Designing for Performance, William Stallings, PHI, 8th Edition.
2. Computer Organization & Embedded Systems, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, & Naraig Majikian, McGraw Hill, 6th Edition
3. Modern Digital Electronics, R. P. Jain, McGraw Hill, 4th Edition
4. Embedded System, Architecture and programming,Rajkamal,TMH,2008

Additional References:

- Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011
- https://www.embeddedrelated.com/showarticle/453.php
- https://www.8051projects.net/wiki/Keil_Embedded_C_Tutorial#Introduction_to_Keil_C

Course Code: RUSCS102

Course Title: PROGRAMMING WITH PYTHON – I

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Interpret the fundamental python syntax and expert in use sequential and looping structure in python structure
CO 2	Understand data storing and processing mechanism using string, List, Dictionary
CO 4	Understand File processing in python
CO 5	Develop python standalone application in real world

DETAILED SYLLABUS

Course Code	Unit	PROGRAMMING WITH PYTHON – I	Lectures
RUSCS102	I	**Why Python?** Reasons for Python as the learner are first programming language. Introduction to the IDLE interpreter (shell) and its documentation. **Building Blocks of Program:** Data, Data Types, Data Binding, Variables, Constants, Declaration, Operations on Data such as assignment, arithmetic, relational, logical operations, dry run, and variables used. **Develop Code using Python:** Features, basic syntax, Writing and executing simple program, Basic Data Types such as numbers, strings, etc Declaring variables, Performing assignments, arithmetic operations, Simple input-output	15 L
	II	**Sequence Control:** Precedence of operators, Type conversion Conditional Statements: if, if-else, nested if –else Looping: for, while, nested loops **Control statements:** Terminating loops, skipping specific conditions Collection Manipulation: declaring strings, string functions, Lists, Tuples, Maps	15 L
	III	**Functions And Modules:** Defining a function, calling a function, Advantages of functions, types of functions, function parameters, Formal parameters, Actual parameters, global and local variables, Anonymous functions, List comprehension Importing module, Creating & exploring modules	15 L

		Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications.	

Course Code	PRACTICAL OF PROGRAMMING WITH PYTHON – I	Credits
RUSCSP102	1. Program based on I/O concepts. 2. Programs based on Control Statement. 3. Program based on Strings. 4. Program based on Tuples. 5. Program based on list. 6. Program based on dictionaries. 7. Program based on Function and anonymous function. 8. Program based on Modules. 9. Programs to read and write files. 10. Programs with iterables and iterators	**1**

References:

1. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress,2nd edition.
2. Practical Programming: An Introduction to Computer Science Using Python, Paul Gries, et al., Pragmatic Bookshelf, 2nd Edition 2014.

Additional References:

- Introduction to Computer Science using Python, Charles Dierbach, Wiley, 2013.
- Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries , Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition 2014
- Programming Languages - Principles and Paradigms, Adesh Pandey, Naros

Course Code: RUSCS103

Course Title: LINUX FUNDAMENTALS

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Understand the concept behind Free and Open Source Software, its use, importance and impact in the society. To explain the open source methodologies and ecosystem to students.
CO 2	Demonstrate various Basic Commands & Advance Commands of Linux in detail.
CO 3	Demonstrate Basic Shell Scripting & Advance Shell Scripting of Linux in detail & to enable students to write programs on Linux platform (Shell scripts/ C programs etc).
CO 4	Articulate Databases in Linux.

DETAILED SYLLABUS

Course Code	Unit	LINUX FUNDAMENTALS	Lectures
RUSCS103	**I**	**FOSS Philosophy:** Introduction to Free and Open Source Software, History of Open Source Software, OSI & FSF, FOSS Advantages / Disadvantages, Economic impact of FOSS, Social impact of FOSS, FOSS in Governments Sectors, FOSS in Education, Software Licensing. **Introduction to Linux:** Linux Architecture Features of Linux, Understanding Linux File system, Linux Distributions, The Linux Console, Linux Desktop, Startup and Shutdown Process, Types of Desktop - X-Windows, KDE, GNOME. **Linux Commands:** General Purpose Utilities, File Handling Utilities, Process Management, Simple Filters, and Filters using Regular Expressions - grep.	15 L
	II	**FILTER COMMAND & EDITORS** Using Advanced Filters: AWK Working with various editors: sed, vi/vim Editor, Gedit, Nano, GNU Emacs, Kwrite, gVim, Bluefish. **SHELL SCRIPTING BASICS** Basic Script Building: Using multiple commands, creating script files, displaying messages, using variables, redirecting input/output, pipes performing math, exiting script. **Using Structured Commands:** working with if-then and if-then-else statements, nesting if's, the test command, compound condition testing, advance if-then feature, the case command. More Structured Commands: The for command, the c-style for command, the while command, the until command, nesting loops, controlling loops.	15 L
	III	**ADVANCED SHELL SCRIPTING** Handling User Input: Command Line Parameters, Special Parameter Variables, Working With Options, Finding your options, Using the getopt command, The more advanced getopts, Getting User Input. Presenting Data: understanding input and output, redirecting output in scripts, redirecting input scripts, creating your own redirections. **Creating Functions**: basic script functions, returning a value, using variables in functions, array variables and functions, function recursion, using functions in the command line. **Writing Scripts for Graphical Desktops**: Creating text menus, doing windows, getting graphics. **Using Databases:** The MySQL database, working with tables, using databases in your scripts.	15 L

Course Code	PRACTICAL OF LINUX FUNDAMENTALS	Credits
RUSCSP103	1. Installation of Ubuntu Linux a. Installing Linux distribution e.g. Ubuntu. b. Customize desktop environment by changing different default options like changing default background, themes, screensavers, changing Screen Resolution.	**1**

	c. Changing time settings and time zone of your system to (or New York Time if you are currently in Indian time). How does the displayed time change? After noting the time change, change the time zone back to your local time zone. d. Installing and Removing Software: Install gcc package. Verify that it runs, and then remove it. 2. Create and publish your own open source project: Write any simple program using your choice of programming language. 3. Create a repository on github and save versions of your project. 4. Executing General Purpose Utility commands. 5. Executing File Handling Utilities. 6. Executing Filter Commands and Regular Expression. 7. Executing Process Management Utilities. 8. Writing shell scripts. 9. Writing C programs using gcc compiler in Linux. 10. Creating GUI application. 11. Working with Databases.	

References:

- Linux Command Line and Shell Scripting, Richard Blum, Christine Bresnahan, 2nd Edition, Wiley India.
- UNIX Concepts & Applications, Sumithbha Das, 4th Edition, Tata McGraw Hill.
- Free/Open Source Software: A General Introduction By Kenneth Wong, Phet Sayo,2004.
- Free Software, Free Society by Richard Stallmann (Second Edition)

Additional References:

- The Linux Documentation Project: http://www.tldp.org/
- Linux kernel Home: http://kernel.org
- The Linux Foundation: http://www.linuxfoundation.org/
- Open Source Database Technologies -
- http://blog.capterra.com/free-database-software/

Course Code: RUSCS104

Course Title: ALGORITHMS AND PROGRAMMING WITH C
Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Critically think ,Assess and choose best solution for solving problem
CO 2	Design Algorithms and Flowcharts for representing Logic
CO 3	Interpret syntax and semantics of C programming
CO 4	Develop modular Programs and Applications of data structures.

CO 5	Use Pointers, Structures and File processing for various applications.

DETAILED SYLLABUS

Course Code	Unit	ALGORITHMS AND PROGRAMMING WITH C	Lectures
RUSCS104	**I**	**Fundamentals of algorithms:** Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures. Different approaches in programming: Procedural approach, Object Oriented approach, Event Driven approach. Structure of C: Header and body, Use of comments, Compilation of a program. Interpreters vs. compilers. Data Concepts: Variables, Constants, data types. 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. **Type conversions:** Automatic and Explicit type conversion. **Iterations:** Control statements for decision making: (i) Branching: if statement, else.. if statement, switch statement. (ii) Looping: while loop, do.While, for loop. (iii) Jump statements: break, continue and goto.	15 L
	II	**Arrays:** (One and multidimensional), declaring array variables, initialization of arrays, accessing array elements. **Strings:** Declaring and initializing String variables, Character and string handling functions **Data Input and Output functions:** Formatted I/O: printf(), scanf(). 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. **Recursion:** Definition, Recursive functions. Storage Classes:Automatic, External, static, RegisterVarable	15 L
	III	**Pointer:** Fundamentals, Pointer variables, Referencing and dereferencing, Pointer Arithmetic, Using Pointers with Arrays, Using Pointers with Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers. **Dynamic Memory Allocation:** malloc(), calloc(), realloc(), free() and sizeof operator. **Structure:** Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures. **Unions:** Defining and working with unions. **File handling:** Different types of files like text and binary, Different types of functions.	15 L

Course Code	PRACTICAL OF ALGORITHMS AND PROGRAMMING WITH C	Credits
RUSCSP104	1. Develop Algorithms and design flow chart along with c program for the given problem 2. Program to understand Basic Data types and and I/O 3. Programs on Operators and Expression 4. Programs on Control Structures 5. programs on functions 6. Programs on Array 7. Programs on Dynamic Memory Allocation 8. Programs on Strings 9. Programs on Structures and Unions	**1**

References:

1. Programming in ANSI C, E Balagurusamy, TMH, 3rd Edition.

Additional References:

- Let Us C, Yashavant P. Kanetkar, BPB Publications

Course Code: RUSCS105

Course Title: DISCRETE MATHEMATICS
Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Analyze discrete objects, starting with relations and partially ordered sets.
CO 2	Interpret recurrence relations, generating function and operations on them.
CO 3	Construct graphs and trees, which are widely used in software.
CO 4	Apply basic knowledge about models of automata theory and the corresponding formal languages.
CO 5	Solve different problems related to Counting principles and relations.

DETAILED SYLLABUS

Course Code	Unit	DISCRETE MATHEMATICS	Lectures
RUSCS105	**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.	15 L

		Relations: Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Diagrams , Maximum and Minimum elements, Lattices **Recurrence Relations:** Definition of recurrence relations, Formulating recurrence relations,solving recurrence relations, Backtracking method, Linear homogeneous recurrence relations with constant coefficients.	
	II	**Counting Principles , Languages and Finite State Machine** **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). **Languages, Grammars and Machines:** Languages , regular Expression and Regular languages, Finite state Automata, grammars, Chomsky hierarchy of type-0, type-1, type-2 and type-3 grammars	15 L
	III	**Graphs and Trees** **Graphs:** Definition and elementary results, Adjacency matrix, pa th matrix, Representing relations using digraphs, Warshall's algorithm- shortest path, Linked representation of a graph, Operations on graph with algorithms, 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	15 L

Course Code	PRACTICAL OF DISCRETE MATHEMATICS	Credits
RUSCSP105	1. Graphs of standard functions such as absolute value function, in verse function, logarithmic and exponential functions, flooring and ceiling functions, trigonometric functions over suitable intervals. 2. Partial ordering sets, Hasse diagram and Lattices. 3. Recurrence relation. 4. Different counting principles. 5. Finite state Automata and Finite state machines. 6. Warshall's Algorithm. 7. Shortest Path algorithms. 8. Operations on graph. 9. Breadth and Depth First search algorithms. 10. Concept of searching, inserting and deleting from binary search trees.	**1**

References:

1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011)
2. Discrete Mathematics, Norman L. Biggs, Clarendon Press, Oxford 1989, Revised Edition.
3. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw- Hill Inc.

Additional References:

- Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition .
- Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
- Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc.
- Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New Delhi.

Course Code: RUSCS106

Course Title: DESCRIPTIVE STATISTICS AND INTRODUCTION TO PROBABILITY
Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Apply basic statistics essential for prospective researchers and professionals to know these basics.
CO 2	Interpret descriptive statistical concepts and its use
CO 3	Understand the probability concept required for Computer Science

DETAILED SYLLABUS

Course Code	Unit	DESCRIPTIVE STATISTICS AND INTRODUCTION TO PROBABILITY	Lectures
RUSCS106	**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, quant iles Real life examples	15 L
	II	**Moments:** raw moments, central moments, relation between raw and central moments	15 L

		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)	
	III	**Probability:** Random experiment, sample space, events types and operations of events **Probability definition:** classical, axiomatic, Elementary Theorems of probability (without proof) • $0 \leq P(A) \leq 1$, • $P(A \quad B) = P(A) + P(B) - P(A \cap B)$ • P (A') = 1 - P(A) • $P(A) \leq P(B)$ if $A \quad B$ Conditional probability, 'Bayes' theorem, independence, Examples on Probability **Standard distributions:** random variable; discrete, continuous, expectation and variance of a random variable, pmf, pdf, cdf, reliability	15 L

Course Code	PRACTICAL OF DESCRIPTIVE STATISTICS AND INTRODUCTION TO PROBABILITY	Credits
RUSCSP106	1. Frequency distribution and data presentation 2. Measures of central tendency 3. Data entry using, functions, c(), scan (), Creating vectors, Mathematical Operations: ** +/-/*/ / ^ , exp, log, log10, etc, creating vector of text type, useful functions: data, frame, matrix operations, seq(), split() etc. 4. Frequency distribution using cut(), table() 5. Data presentation 6. Summary Statistics (measures of central tendency, dispersion) 7. Measures of skewness and kurtosis 8. Correlation and regression 9. Probability 10. Conditional probability	**1**

References:

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

Additional References:

- A First course in probability, Ross, S.M. (2006):, Pearson, 6th Edition.
- common statistical tests, Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999), Satyajeet Prakashan, Pune

- Fundamentals of Mathematical Statistics, Gupta, S.C. and Kapoor, V.K. (1987), S. Chand and Sons, New Delhi
- Applied Statistics, Gupta, S.C. and Kapoor, V.K. (1999), S˙Chand and Son's, New Delhi
- Planning and Analysis of Experiments, Montgomery, D.C. (2001): wiley.

Course Code: RUSCS107

Course Title: SOFT SKILLS DEVELOPMENT
Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Interpret about various aspects of soft skills and learn ways to develop personality
CO 2	Understand the importance and type of communication in a personal and professional environment.
CO 3	Develop insight into much needed technical and non-technical qualities in career planning
CO 4	Develop Leadership, team building, decision making and stress management skills
CO 5	Develop professional, social and academic skills to harness hidden strengths, capabilities and knowledge
CO 6	Understand various issues in personal and professional communication and learn to overcome them.

DETAILED SYLLABUS

Course Code	Unit	SOFT SKILLS DEVELOPMENT	Lectures
RUSCS107	**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. **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.	15 L

	II	**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. **Job** **Interviews:** Introduction, Importance of Resume, Definition of In terview, 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 Arrangeme nt 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	15 L
	III	**Professional Skills Creativity at Workplace:** Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method **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 **Capacity Building:** Learn, Unlearn and Relearn: Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building 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 Decisi on 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	15 L

References:

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

Additional References:

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

MODALITY OF ASSESSMENT

Theory exam total marks: 100 Marks

Theory Examination Pattern:

A) Internal Assessment - 40% :40 marks.

Sr No	Evaluation type	Marks
1	It will be conducted either using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment)	20
2	Project (group of 5 students)/Tutorial/Quizzes/Assignment	20
3	Total	40

B) External examination - 60 % : 60 marks

Semester End Theory Examination:

1. Duration - These examinations shall be of **2 Hrs** duration.
2. Theory Question Paper Pattern:

All Questions are Compulsory			
Questions	**Options**	**Based On**	**Marks**
Q1	Any 3 out of 4	Unit I, II, & III	15
Q2	Any 3 out of 4	Unit I	15
Q3	Any 3 out of 4	Unit II	15
Q4	Any 3 out of 4	Unit III	15
Total			60

- All questions shall be compulsory with internal choice within the questions.

Practical exam total marks: 50 Marks

Practical Examination Pattern:

(A) Internal Examination: Internal Practical 40% - 20 Marks

10 Marks - Individual Practical Implementation & Performance

- Each student will maintain an e-journal. After every practical students will upload his practicals in the form of documents along with the screen shots of output on online portal (Moodle/Google site/any LMS).

<u>10 Marks</u> –Design and implement innovative application of the technology

Particulars	**Marks**
Individual Practical Implementation & Performance	10
Design and implement innovative application of the technology	10
Total	20

(B) External (Semester end practical examination): <u>60% - 30 Marks</u>
<u>30 Marks Practical Questions:</u>

- Students have to acquire atleast 40% marks in each paper individually.

Particulars	**Practical**
Laboratory work	30
Total	**30**

<u>PASSING CRITERIA 40%: -</u> Student has to acquire minimum of 40% marks each course (Theory and Practical) both.

Overall Examination & Marks Distribution Pattern

Semester I

Course	**101, 102, 103, 104, 105, 106, 107**		
	Internal	**External**	**Grand Total**
Theory	**40**	**60**	**700 (7 Papers)**
Practicals	**20**	**30**	**300 (6 Papers)**
Individual Semester Total			**1000**

Course Code: RUSCS201

Course Title: DATABASE MANAGEMENT SYSTEMS
Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Know about DBMS with respect to the relational model, to specify the functional and data requirements for a typical database application
CO 2	Understand creation, manipulation and querying of data in databases.
CO 3	Evaluate business information problems and find the requirements of a problem in terms of data.
CO 4	Design the database schema with the use of appropriate data types for storage of data in the database.
CO 5	Create, manipulate, query and back up the databases.

DETAILED SYLLABUS

Course Code	Unit	DATABASE MANAGEMENT SYSTEMS	Lectures
RUSCS201	**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) **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 **ER to Table-** Entity to Table, Relationship to tables with and without key constraints	15 L
	II	**Schema Refinement And Normal Forms:** Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition. **Relational Algebra -** operations (selection, projection, set operations union, intersection, difference, cross product, Joins – conditional, equi join and natural joins, division)	15 L

		DDL Statements - Creating Databases, Using Databases, data types, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables, Backing Up and Restoring databases **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	
	III	**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 **Database Protection-** Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control **DCL Statements -**creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges	15 L

Course Code	PRACTICAL OF DATABASE MANAGEMENT SYSTEMS	Credits
RUSCSP201	1. For given scenario ● Draw an 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: a. Viewing all databases b. Creating a Database c. Viewing all Tables in a Database d. Creating Tables (With and Without Constraints) e. Inserting/Updating/Deleting Records in a Table f. Saving (Commit) and Undoing (rollback) 4. Perform the following: a. a. Altering a Table b. Dropping/Truncating/Renaming Tables c. Backing up / Restoring a Database 5. Perform the following: d. Simple Queries e. Simple Queries with Aggregate functions f. Queries with Aggregate functions (group by and having clause) 6. Queries involving g. Date Functions h. String Functions i. Math Functions 7. Join Queries j. Inner Join k. Outer Join	**1**

	8. Subqueries l. With IN clause m. With EXISTS clause 9. Views n. Creating Views (with and without check option) o. Dropping views p. Selecting from a view 10. DCL statements q. Granting and revoking permissions	

References:

1. Fundamentals of Database Systems, Ramez Elmasri & Shamkant B.Navathe, Pearson Education, 6th Edition, 2010.
2. Database Management Systems, Ramakrishnam, Gehrke, McGraw-Hill, 2007.
3. Murach's MySQL, Joel Murach, Murach, 2012.

Additional References:

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

Course Code: RUSCS202

Course Title: PROGRAMMING WITH PYTHON - II

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Articulate OOPS concept such as Inheritance, polymorphism in python
CO 2	Explain exception handling in python
CO 3	Identify commonly used functions in Regular expression for efficiency
CO 4	Develop GUI application with database
CO 5	Illustrate Network programming in python

DETAILED SYLLABUS

Course Code	Unit	PROGRAMMING WITH PYTHON - II	Lectures
RUSCS202	**I**	**Fundamental ideas of OOP**: encapsulation, inheritance, abstraction, polymorphism, Classes, Objects in python **Exception Handling:** What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise. **Regular Expressions:** Concept of regular expression, various types of regular expressions, using match function	15 L
	II	**GUI Programming in Python :** 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:** frame, label, button, checkbutton, entry, listbox, message, radiobutton, text, spinbox etc	15 L
	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

Course Code	PRACTICAL OF PROGRAMMING WITH PYTHON - II	Credits
RUSCSP202	1. Programs based on File processing 2. Programs based on Exception handling 3. Programs based on regular expressions 4. Programs based on Draw shapes and animations 5. Programs based on GUI Controls 6. Programs based on Database Connectivity 7. Programs based on networking 8. Programs based on send email and read content	**1**

References:

1. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries , Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition 2014

Additional References:

- Beginning Python: Using Python 2.6 and Python 3, James Payne , Wiley India, 2010.
- MySQL for Python: Database Access Made Easy, A. Lukaszewski, Pact Publisher, 2010.

Course Code: RUSCS203

Course Title: LINUX SERVER ADMINISTRATION

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Understand Linux Administration Duties & the Linux Administrative commands to manage the resource on Linux machines
CO 2	Understand the install, manage and maintain the softwares in Linux Server.
CO 3	Execute all the Linux Commands required for user related activities in Linux.
CO 4	Develop various scripts required for the automation of few services in Linux.
CO 5	Apply various Intranet Services & Internet Services on Linux Servers.

DETAILED SYLLABUS

Course Code	Unit	LINUX SERVER ADMINISTRATION	Lectures
RUSCS203	**I**	**SYSTEM ADMINISTRATION Duties of the System Administrator:** the Linux system administrator, installing and configuring servers, installing and configuring application software, creating and maintaining user accounts, backing up and restoring files, monitoring and tuning performance, configuring a secure system, using tools to monitor security. **Planning the Network:** deciding the kind of network, planning and implementing security, planning for recovery from disasters, **Red Hat Linux File System:** understanding file system structure, using file system commands, working with Linux supported file system, Linux disk management. **Red Hat System Configuration:** examining the system configuration file, examining the network configuration files, managing the init scripts.	15 L
	II	**Single-Host Administration:** Managing Users and Groups, Booting and shutting down processes, File Systems, Core System Services, Process of configuring, compiling, Linux Kernel. **Networking and Security:** TCP/IP for System Administrators, Process of configuring, Linux Firewall (Netfilter), System and network security.	15 L
	III	**Internet Services:** Domain Name System (DNS), File Transfer Protocol (FTP), Apache web server, Simple Mail Transfer Protocol (SMTP), Post Office Protocol and Internet Mail Access Protocol (POP and IMAP), Secure Shell (SSH), Network Authentication,OpenLDAP Server, Samba and LDAP, Networ	15 L

		k authentication system (Kerberos), Domain Name Service (DNS), Security **Intranet Services**: Network File System (NFS), Samba, Distributed File Systems (DFS), Network Information Service (NIS), Lightweight Directory Access Protocol (LDAP),Dynamic Host Configuration Protocol (DHCP), MySQL, LAMP Applications File Servers, Email Services, Chat Applications, Virtual Private Networking.	

Course Code	PRACTICAL OF LINUX SERVER ADMINISTRATION	Credits
RUSCSP203	1. Installation of Red HAT Linux operating system. a. Partitioning drives b. Configuring boot loader (GRUB/LILO) c. Network configuration d. Setting time zones e. Creating password and user accounts f. Shutting down 2. Linux system administration a. Becoming super user b. Temporarily changing user identity with su command c. Using graphical administrative tools d. Administrative commands e. Administrative configuration files 3. Configuring Network: a. Get the IP address of your machine using ifconfig. b. If IP is not set, then assign an IP address according to your network settings. c. Get the hostname of your machine. d. Use ping to check the network connectivity to remote machines. e. Use telnet/ssh to connect to remote machines and learn the difference between the two. f. Troubleshooting network using traceroute, ping, route commands. 4. Configuring samba Server. 5. Install DHCP Server 6. Configure NTP Server (NTPd), Install and Configure NTPd, 7. Configure NTP Client (Ubuntu and Windows) 8. SSH Server : Password Authentication Configure SSH Server to manage a server from the remote computer, SSH Client : (Ubuntu and Windows 9. Install DNS Server BIND, Configure DNS server which resolves domain name or IP address, Install BIND 9, Configure BIND, Limit ranges you allow to access if needed. 10. Configure NIS Server in order to share users' accounts in your local networks, Configure NIS Client to bind NIS Server. 11. Configure LDAP Server, Configure LDAP Server in order to share users' accounts in your local networks, Add LDAP User Accounts in the	**1**

	OpenLDAP Server, Configure LDAP Client in order to share users' accounts in your local networks.	

References:

1. RedHat Linux Networking & System Administration, Terry Collings, Kurt Wall, E-Book.
2. Linux Administration A Beginners Guide, Wale Soyinka

Course Code: RUSCS204

Course Title: DATA STRUCTURES
Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Explore and understand the concepts of Data Structures and its significance in programming.
CO 2	Provide a holistic approach to design, use and implement abstract data types.
CO 3	Understand the commonly used data structures and various forms of its implementation for different applications using Python.
CO 4	Apply various Data structures algorithms to real time application.

DETAILED SYLLABUS

Course Code	Unit	DATA STRUCTURES	Lectures
RUSCS204	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 Multiarrays, Application **Algorithm Analysis**: seven standard functions, Complexity Analysis-Big-O Notation, Evaluating Python Code, Evaluating Python List, Amortized Cost, Evaluating Set ADT, Application. **Searching and Sorting**: Searching-Linear Search, Binary Search, Sorting-Bubble, Selection and Insertion Sort, Working with Sorted Lists-Maintaining Sorted List, Maintaining sorted Lists.	15 L
	II	**Linked Structures**: Introduction, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, Bag ADT-Linked List Implementation. Comparing	15 L

		Implementations, Linked List Iterators, More Ways to Build l inked Lists, Applications-Polynomials **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 Advanced Linked List: Doubly Linked Lists-Organization and Operation, Circular Linked List-Organization and Operation, Multi Lists	
	III	**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 **Binary Trees:** Tree Structure, Binary Tree-Properties, Implementation and Traversals, Expression Trees, Heaps and Heap sort, Search Trees	15 L

Course Code	**PRACTICAL OF DATA STRUCTURES**	**Credits**
RUSCSP204	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 Evaluation 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. 10. Implement Binary Tree and its traversals. 11. Recursive implementation of a. Factorial b. Fibonacci c. Tower of Hanoi	**1**

References:

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

Additional References:

- Data Structure and Algorithmic Thinking with Python- Narasimha Karumanchi, 2015, Careermonk Publications
- Fundamentals of Python: Data Structures, Kenneth Lambert, Delmar Cengage Learning

Course Code: RUSCS205

Course Title: CALCULUS

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Discover the Mathematical concepts like limit, continuity, derivative, integration of functions
CO 2	Apply the concepts to real world problems.
CO 3	Formulate a problem through Mathematical modeling and simulation.
CO 4	Evaluate the problems related to integration, derivatives and its applications.

DETAILED SYLLABUS

Course Code	Unit	CALCULUS	Lectures
RUSCS205	**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, Newton's Method	15 L
	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 and Applications.	15 L
	III	**Partial Derivatives and its Applications:**	15 L

		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 Variables.	

Course Code	PRACTICAL OF CALCULUS	Credits
RUSCSP205	Implement the following practical in R/Python. 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	**1**

References:

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

Additional References:

- Calculus and analytic geometry, George B Thomas, Ross L Finney, Addison Wesley, 1995, 9th edition.
- Calculus: Early Transcendentals, James Stewart, Brooks Cole, 2015, 8th Edition.
- Calculus, Ron Larson, Bruce H. Edwards, Cengage Learning, 2013, 10th Edition.
- Thomas' Calculus, George B. Thomas, Maurice D. Weir, Joel R. Hass, Pearson, 2014, 13th Edition

Course Code: RUSCS206

Course Title: STATISTICAL METHODS

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Apply basic statistics to real world applications.
CO 2	Understand hypothesis testing
CO 3	Formulate One & Two way ANOVA
CO 4	Evaluate real world problems using linear programming.

DETAILED SYLLABUS

Course Code	Unit	STATISTICAL METHODS	Lectures
RUSCS206	**I**	**Statistical Models :** Useful statistical model, Discrete distribution, Continuous distribution, Binomial, Normal, chi-square, t, F. Examples **Non-Parametric Tests:** need of non-parametric tests, sign test, Wilicoxon's signed rank test, run test, Kruskal-Walis tests.	15 L
	II	**Hypothesis Testing:** one sided, two sided hypothesis, critical region, p-value, tests based on t, Normal and F, confidence intervals. **Analysis Of Variance:** one-way, two-way analysis of variance	15 L
	III	**Linear Programming model:** Formulation & solving linear programming problem using Graphical method for two variable problems, Simplex methods, Dual Simplex Method.	15 L

Course Code	PRACTICAL OF STATISTICAL METHODS	Credits/ Lectures
RUSCSP206	1. Plotting pdf, cdf, pmf, for discrete and continuous distribution 2. Problems based on discrete & continuous distribution. 3. t test, normal test, F test 4. Analysis of Variance 5. Non parametric tests- I 6. Non- Parametric tests - II 7. Post-hoc analysis of one-way analysis 8. LPP for maximization /minimization of an objective function and graphical representation of feasible solutions. 9. Simple Simplex 10. Dual Simplex	**1**

References:

1. Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Trivedi, K.S.(2009), Prentice Hall of India, New Delhi.
2. Operation Research - An Introduction by H.A. Taha,8th edition.
3. Operations Research Theory and Applications by J.K. Sharma,5th edition.

Additional References:

- Operations Research by P.K . Gupta, Hira S. Chand
- Optimization Methods by Mital K.V
- A First course in probability, Ross, S.M. (2006), Pearson, 6th Edition.
- Common statistical tests, Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999), Satyajeet Prakashan, Pune.
- Fundamentals of Mathematical Statistics, Gupta, S.C. and Kapoor, V.K. (2002), S. Chand and Sons, New Delhi.
- Applied Statistics, Gupta, S.C. and Kapoor, V.K., S. Chand and Son's, New Delhi, 4th Edition.
- Planning and Analysis of Experiments, Montgomery, D.C. (2001), Wiley

Course Code: RUSCS207

Course Title: GREEN TECHNOLOGIES

Academic year 2021-22

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Understanding the concept of Green Computing and Green IT infrastructure for making computing and information system environment sustainable.
CO 2	Encouraging optimized software and hardware designs for development of Green IT Storage, Communication and Services.
CO 3	Interpret the highlight useful approaches to embrace green IT initiatives.
CO 4	Learning about green IT can be achieved in and by hardware, software, network communication and data center operations.
CO 5	Understanding the strategies, frameworks, processes and management of green IT

DETAILED SYLLABUS

Course Code	Unit	GREEN TECHNOLOGIES	Lectures
RUSCS207	**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-	15 L

		Labelling 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 Pra ctices, Sustainable Software, Software Sustainability Attributes, Software Susta inability Metrics, Sustainable Software Methodology, Defining Actions	
	II	**Green Data Centers**: Data Centers 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 Strategy, Steps in Developing a Green IT Strategy, Metrics and Measurements in Green Strategies	15 L
	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. **Green Enterprises and the Role of IT**: Introduction, Organizational and Enterprise Greening, Information Systems in Greening Enterprises, **Greening the Enterprise**: IT Usage and Hardware, Inter-organizational Enterprise Activities and Green Issues	15 L

References:

1. Harnessing Green IT: Principles and Practices, San Murugesan, G. R. Ganadharan, Wiley & IEEE,Reprint 2013.

Additional References:

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

MODALITY OF ASSESSMENT

Theory exam total marks: 100 Marks

Theory Examination Pattern:

A) Internal Assessment - 40% :40 marks.

Sr No	Evaluation type	Marks
1	It will be conducted either using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment)	20
2	Project (group of 5 students)/Tutorial/Quizzes/Assignment	20
3	Total	40

B) External examination - 60 % : 60 marks

Semester End Theory Examination:

4. Duration - These examinations shall be of **2 Hrs** duration.
5. Theory Question Paper Pattern:

All Questions are Compulsory			
Questions	**Options**	**Based On**	**Marks**
Q1	Any 3 out of 4	Unit I, II, & III	15
Q2	Any 3 out of 4	Unit I	15
Q3	Any 3 out of 4	Unit II	15
Q4	Any 3 out of 4	Unit III	15
Total			60

- All questions shall be compulsory with internal choice within the questions.

Practical exam total marks: 50 Marks

Practical Examination Pattern:

A) Internal Examination: Internal Practical 40% - 20 Marks

10 Marks - Individual Practical Implementation & Performance

- Each student will maintain an e-journal. After every practical students will upload his practicals in the form of documents along with the screen shots of output on online portal (Moodle/Google site/any LMS).

10 Marks –Design and implement innovative application of the technology

Particulars	Marks
Individual Practical Implementation & Performance	10
Design and implement innovative application of the technology	10
Total	20

(B) External (Semester end practical examination): 60% - 30 Marks

30 Marks Practical Questions:

- Students have to acquire at least 40% marks in each paper individually.

Particulars	Practical
Laboratory work	30
Total	**30**

PASSING CRITERIA 40%: - **Student has to acquire minimum of 40% marks each course (Theory and Practical) both.**

Overall Examination & Marks Distribution Pattern

Semester II

Course	201, 202, 203, 204, 205, 206, 207		
	Internal	External	Grand Total
Theory	40	60	700 (7 Papers)
Practicals	20	30	300 (6 Papers)
Individual Semester Total			1000