S. P. Mandali's

Ramnarain Ruia Autonomous College

(Affiliated to University of Mumbai)

Syllabus for FY, SY, TY

Program: B.Sc (Computer Science)

Program Code: Computer Science (RUSCS)

(Credit Based Semester and Grading System
for academic year 2020–2021)

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 work place 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	-----	-------	-------

S.Y.B.Sc (Computer Science)							
SEMESTER – III (THEORY)					SEMESTER – III (PRACTICALS)		
YEAR	SEM	COURSE CODE	COURSE TITLE	CREDITS	COURSE CODE	COURSE TITLE	CREDITS
S.Y.Bsc	III	RUSCS301	Theory of Computation	2	-------	-----	------
S.Y.Bsc	III	RUSCS302	Core JAVA	2	RUSCSP302	Practical of Core JAVA	1
S.Y.Bsc	III	RUSCS303	Operating System	2	RUSCSP303	Practical of Operating System	1
S.Y.Bsc	III	RUSCS304	Database	2	RUSCSP304	Practical of Database	1

			Management Systems			Management Systems	
S.Y.Bsc	**III**	**RUSCS305**	Combinatorics and Graph Theory	2	**RUSCSP305**	Practical of Combinatorics and Graph Theory	1
S.Y.Bsc	**III**	**RUSCS306**	Physical Computing and IoT Programming	2	**RUSCSP306**	Practical of Physical Computing and IoT Programming	1
S.Y.Bsc	**III**	**RUSCS307**	Skill Enhancement: Web Programming	2	**RUSCSP307**	Practical of Skill Enhancement: Web Programming	1

S.Y.B.Sc (Computer Science)							
SEMESTER – IV (THEORY)					**SEMESTER – IV (PRACTICALS)**		
YEAR	**SEM**	**COURSE CODE**	**COURSE TITLE**	**CREDITS**	**COURSE CODE**	**COURSE TITLE**	**CREDITS**
S.Y.Bsc	**IV**	**RUSCS401**	Fundamentals of Algorithms	2	**RUSCSP401**	Practical of Fundamentals of Algorithms	1
S.Y.Bsc	**IV**	**RUSCS402**	Advanced JAVA	2	**RUSCSP402**	Practical of Advanced JAVA	1
S.Y.Bsc	**IV**	**RUSCS403**	Computer Networks	2	**RUSCSP403**	Practical of Computer Networks	1
S.Y.Bsc	**IV**	**RUSCS404**	Software Engineering	2	-----	-----	-------
S.Y.Bsc	**IV**	**RUSCS405**	Linear Algebra	2	**RUSCSP405**	Practical of Linear Algebra	1
S.Y.Bsc	**IV**	**RUSCS406**	.NET Technologies	2	**RUSCSP406**	Practical of .NET Technologies	1
S.Y.Bsc	**IV**	**RUSCS407**	Skill Enhancement: Android Developer Fundamentals	2	**RUSCSP407**	Practical of Skill Enhancement: Android Developer Fundamentals	1

T.Y.B.Sc (Computer Science)							
SEMESTER – V (THEORY)					SEMESTER – V (PRACTICALS)		
YEAR	SEM	COURSE CODE	COURSE TITLE	CREDITS	COURSE CODE	COURSE TITLE	CREDITS
T.Y.Bsc	V	RUSCS501	Artificial Intelligence	3	RUSCSP501	Practical of Artificial Intelligence	1
T.Y.Bsc	V	RUSCS502	Software Testing and Quality Assurance	3	RUSCSP502	Practical of Software Testing and Quality Assurance	1
T.Y.Bsc	V	RUSCS503	Information and Network Security	3	RUSCSP503	Practical of Information and Network Security	1
T.Y.Bsc	V	RUSCS504	Web Services	3	RUSCSP504	Practical of Web Services	1
T.Y.Bsc	V	RUSCS505	Skill Enhancement : Ethical Hacking	2	RUSCSP505	Practical of Skill Enhancement :Ethical Hacking	1
T.Y.Bsc	V	-----	----	----	RUSCSP506	Practical of Advanced Web Programming	1

T.Y.B.Sc (Computer Science)							
SEMESTER – VI (THEORY)					SEMESTER – VI (PRACTICALS)		
YEAR	SEM	COURSE CODE	COURSE TITLE	CREDITS	COURSE CODE	COURSE TITLE	CREDITS
T.Y.Bsc	VI	RUSCS601	Cloud Computing	3	RUSCSP601	Practical of Cloud Computing	1
T.Y.Bsc	VI	RUSCS602	Cyber Forensics	3	RUSCSP602	Practical of Cyber Forensics	1
T.Y.Bsc	VI	RUSCS603	Information Retrieval	3	RUSCSP603	Practical of Information Retrieval	1
T.Y.Bsc	VI	RUSCS604	Data Science	3	RUSCSP604	Practical of Data Science	1

T.Y.Bsc	VI	**RUSCS605**	Skill Enhancement: Optimization Techniques	2	**RUSCSP605**	Project Implementation	2

Course Code: RUSCS101

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

Academic year 2020-21

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,	15 L

		assembly language, stacks, subroutines, additional instructions, CISC instruction sets, RISC & CISC styles **Basic Processing Unit -** 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).	**1**

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.
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 2020-21

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 **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.	15 L

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.	**1**

	10. Programs with iterables and iterators	

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, Narosa

Course Code: RUSCS103

Course Title: LINUX FUNDAMENTALS

Academic year 2020-21

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.	15 L

		Linux Commands: General Purpose Utilities, File Handling Utilities, Process Management, Simple Filters, and Filters using Regular Expressions - grep.	
	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.
	   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.
		1

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 2020-21

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.	15 L

		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.	
	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 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Analyse 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. **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.	15 L
	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	15 L

		(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	
	III	**Graphs and Trees** **Graphs:** Definition and elementary results, Adjacency matrix, path 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, inverse 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 2020-21

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, quantiles Real life examples	15 L
	II	**Moments:** 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)	15 L
	III	**Probability:** Random experiment, sample space, events types and operations of events **Probability definition:** classical, axiomatic, Elementary Theorems of probability (without proof) • $0 \le P(A) \le 1$, • $P(A\ B) = P(A) + P(B) - P(A \cap B)$ • $P(A') = 1 - P(A)$	15 L

| | | • P(A) ≤ P(B) if A 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 | |

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 2020-21

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 Interview, Background Information, Types of Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, Changes in the Interview Process, FAQ During Interviews. **Group Discussion:** Introduction, Ambience/Seating Arrangement for Group Discussion, Importance of Group Discussions, Difference between Group Discussion, Panel Discussion and Debate, Traits, Types of Group Discussions, topic based and Case based Group Discussion, Individual Traits	15 L
	III	**Professional Skills Creativity at Workplace:** Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at	15 L

		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 Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts	
		Stress and Time Management: Stress, Sources of Stress, Ways to Cope with Stress	

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).

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 atleast 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 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 2020-21

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
		1

7. Join Queries
    j.   Inner Join
    k. Outer Join
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 2020-21

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	15 L

		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	
	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 2020-21

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, basic network Configuration, 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, Network 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.	15 L

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 OpenLDAP Server, Configure LDAP Client in order to share users' accounts in your local networks.
		1

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 2020-21

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 Implementations, Linked List Iterators, More Ways to Build linked 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,	15 L

29

	III	**Recursion**: Recursive Functions, Properties of Recursion, Its working, Recursive Applications	15 L
		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	

Circular Linked List-Organization and Operation, Multi Lists (appears as top row of the first table)

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 2020-21

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:** 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.	15 L

Course Code	PRACTICAL OF CALCULUS	Credits
RUSCSP205	Implement the following practical in R/Python. 1. Continuity of functions; Derivative of functions	**1**

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

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 2020-21

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	15 L

		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.	
	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 2020-21

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-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 Practices, Sustainable Software, Software Sustainability Attributes, Software Sustainability Metrics, Sustainable Software Methodology, Defining Actions	15 L
	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,	15 L

		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	
	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	

Course Code: RUSCS301

Course Title: THEORY OF COMPUTATION

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION
	After Completing this course student will be able to :
CO 1	Explain Grammar and Languages
CO 2	Summarize on Automata theory and its application in Language Design
CO 3	Construct Turing Machines and Pushdown Automata.
CO 4	Describe Linear Bounded Automata and its applications

DETAILED SYLLABUS

Course Code	Unit	THEORY OF COMPUTATION	Lectures
RUSCS301	**I**	**Automata Theory:** Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata. **Formal Languages:** Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata.	15 L
	II	**Regular Sets and Regular Grammar:** Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar. **Context Free Languages:** Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG. **Pushdown Automata:** Definitions, Acceptance by PDA, PDA and CFG.	15 L
	III	**Linear Bounded Automata:** The Linear Bounded Automata Model, Linear Bounded Automata and Languages. **Turing Machines:** Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine. **Undecidability:** The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable	15 L

		Problems.	
		Tutorials : 1. Problems on generating languages for given simple grammar 2. Problems on DFA and NDFA equivalence 3. Problems on generating Regular Expressions 4. Problems on drawing transition state diagrams for Regular Expressions 5. Problems on Regular Sets and Regular Grammar 6. Problems on Ambiguity of Grammar 7. Problems on working with PDA 8. Problems on working with Turing Machines 9. Problems on generating derivation trees 10. Problems on Linear Bound Automata/Universal Turing Machine	

References:
1. Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI,3rd Edition
2. Introduction to Computer Theory, Daniel Cohen, Wiley,2nd Edition
3. Introductory Theory of Computer Science, E.V. Krishnamurthy, Affiliated East-West Press, 2nd Edition.

Additional References:
1. Theory of Computation, Kavi Mahesh, Wiley India
2. Elements of The Theory of Computation, Lewis, Papadimitriou, PHI
3. Introduction to Languages and the Theory of Computation, John E Martin, McGraw-Hill Education
4. Introduction to Theory of Computation, Michel Sipser, Thomson

Course Code: RUSCS302

Course Title: CORE JAVA
Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Design java applications using basic concepts.
CO 2	Understand the various Object oriented programming concepts using Java
CO 3	Develop and implement client-side and server-side programs.
CO 4	Develop and implement exception handling, multithreading.
CO 5	Knowledge and implementation of the swing package

DETAILED SYLLABUS

Course Code	Unit	CORE JAVA	Lectures
RUSCS302	I	**The Java Language:** Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays. **OOPS:** Introduction, Class, Object, Static Keywords, Constructors, this Keyword, Inheritance, super Keyword, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces. **String Manipulations:** String, String Buffer, String Tokenizer. **Packages:** Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages, Access specifiers	15 L
	II	**Exception Handling:** Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples **Multithreading**: Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, Wait() notify() notify all() methods **I/O Streams:** Introduction, Byte-oriented streams, Character-oriented streams, File, Random access File, Serialization **Networking:** Introduction, Socket, Server socket, Client – Server Communication	15 L
	III	**Wrapper Classes:** Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes **Collection Framework:** Introduction, util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes **Inner Classes:** Introduction, Member inner class, Static inner class, Local inner class, Anonymous inner class **Swing:** Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: Jlabel, JTextField and JPasswordField, JTextAres, JButton, JCheckBox, JRadioButton, JComboBox and JList	15 L

Course Code	PRACTICAL OF CORE JAVA	Credits
RUSCSP302	1. Accept integer values for a, b and c which are coefficients of quadratic equations. Find the solution of the quadratic equation. 2. Accept two n x m matrices. Write a Java program to find addition of these matrices. 3. Accept n strings. Sort names in ascending order. 4. Create a package: Animals. In the package animals create an interface Animal with suitable behaviors. Implement the interface Animal in the same package animals. 5. Demonstrate Java inheritance using extended keywords. 6. Demonstrate method overloading and method overriding in Java. 7. Demonstrate creating your own exception in Java. 8. Using various swing components, design Java applications to	**1**

	accept a student's resume. (Design form)	
	9. Write a Java List example and demonstrate methods of Java List interface.	
	10. Design a simple calculator GUI application using swing components.	

References:
1. Herbert Schildt, Java The Complete Reference, Ninth Edition, McGraw-Hill Education, 2014

Additional References:
- E. Balagurusamy, Programming with Java, Tata McGraw-Hill Education India, 2014
- Programming in JAVA, 2nd Ed, Sachin Malhotra & Saurabh Choudhary, Oxford Press
- The Java Tutorials: http://docs.oracle.com/javase/tutorial/

Course Code: RUSCS303

Course Title: OPERATING SYSTEM

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION After Completing this course student will be able to :
CO 1	Explain various components of computer hardware and how processes are working in operating systems.
CO 2	Discuss the structure of the operating system, its functions and algorithms.
CO 3	Understanding the working of operating system, its structures and functioning
CO 4	Compare various algorithms used in operating systems.

DETAILED SYLLABUS

Course Code	Unit	OPERATING SYSTEM	Lectures
RUSCS303	I	**Introduction and Operating-Systems Structures:** Definition of Operating system, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments **Operating-System Structures:** Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure **Processes:** Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication **Threads:** Overview, Multicore Programming, Multithreading	15 L

		Models	
	II	**Process Synchronization:** General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors **CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	15 L
	III	**Main Memory:** Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table **Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing **Mass-Storage Structure:** Overview, Disk Structure, Disk Scheduling, Disk Management **File-System Interface:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing **File-System Implementation:** File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management	15 L

Course Code	Practical Of OPERATING SYSTEM	Credits
RUSCSP303	1. Write a Program to implement First Come First Serve(FCFS) Scheduling. 2. Write a Program to implement Shortest Job First (SJF) Scheduling. 3. Write a Program to implement Priority based Scheduling. 4. Write a Program to implement Round Robin. 5. Write a Program to implement FIFO page replacement algorithm. 6. Write a Program to implement LRU Page replacement algorithm. 7. Write a Program to implement an Optimal page replacement algorithm 8. Write a Program to implement Dinning Philosophers. 9. Write a Program to implement DeadLock Detection. 10. Write a Program to implement the Worst FIT Algorithm.	**1**

References:
1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley,8th Edition
Additional References:
- Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill
- Naresh Chauhan, Principles of Operating Systems, Oxford Press
- Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016

Course Code: RUSCS304

Course Title: DATABASE MANAGEMENT SYSTEMS

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION After Completing this course student will be able to :
CO 1	Develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage.
CO 2	Learn about using PL/SQL for data management
CO 3	Develop and implement Master concepts of stored procedure and triggers and its use
CO 4	Understand concepts and implementations of Exception handling
CO 5	Learn and understand Database Programming Paradigms

DETAILED SYLLABUS

Course Code	Unit	DATABASE MANAGEMENT SYSTEMS	Lectures
RUSCS304	**I**	**Fundamentals of PL/SQL:** Defining variables and constants, PL/SQL expressions and comparisons: Logical Operators, Boolean Expressions, CASE Expressions Handling, Null Values in Comparisons and Conditional Statements, PL/SQL Datatypes: Number Types, Character Types, Boolean Type, Date time and Interval Types. The %TYPE Attribute ,The %ROWTYPE Attribute **Overview of PL/SQL Control Structures:** Conditional Control: IF and CASE Statements, IF-THEN Statement, IF-THEN-ELSE Statement, IF THEN-ELSEIF Statement, CASE Statement, Iterative Control: LOOP and EXIT Statements, WHILE-LOOP, FOR-LOOP, Sequential Control: GOTO and NULL Statements, Continue **Sequences:** creating sequences, referencing, altering and dropping a sequence	15 L
	II	**Stored Procedures & Functions:** Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures. Create a Simple Function, Execute a Simple Function, recursive function. **Triggers:** Concept of triggers, Implementing triggers – creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting and modifying triggers, and enforcing data integrity through triggers. **Cursors:** Concept of a cursor, types of cursors: implicit cursors;	15 L

		explicit cursor, Cursor for loops, Cursor variables, parameterized cursors, nested cursors, FOR UPDATE Clause and WHERE CURRENT Clause	
	III	**Exception Handling:** Understand Exceptions, Handle Exceptions with PL/SQL, Trap Predefined Oracle Server Errors, Trap Non-Predefined Oracle Server Errors, Trap User-Defined Exceptions, Propagate Exceptions, RAISE_APPLICATION_ERROR Procedure. **Query evaluation:** System Catalog, Evaluation of relational operators like selection, projection, join and set, introduction to query optimization**. File Organization and Indexing:** Cluster, Primary and secondary indexing, Index data structure: hash and Tree based indexing, Comparison of file organization: cost model, Heap files, sorted files, clustered files. Creating, dropping and maintaining indexes	15 L

Course Code	PRACTICAL OF DATABASE MANAGEMENT SYSTEMS	**Credits**

RUSCSP304	1. Writing PL/SQL Blocks with basic programming constructs of sequential statements
	     a. CONSTANT
	     b. NOT NULL
	     c. DEFAULT
	     d. %TYPE and % ROWTYPE Attribute.
	 2. Writing PL/SQL Blocks with basic programming constructs by including following:
	     a. If...then...Else, IF...ELSEIF...ELSE... END IF
	     b. Case statement
	 3. Writing PL/SQL Blocks with basic programming constructs by including following:
	     a. While-loop Statements
	     b. For-loop Statements
	     c. Unconstrained loops
	 4. Writing PL/SQL Blocks with basic programming constructs by including Sequences:
	     a. Creating simple Sequences with clauses like START WITH, INCREMENT
	   BY, MAXVALUE, MINVALUE, CYCLE | NOCYCLE, CACHE | NOCACHE, ORDER | NOORECER.
	     b. Creating and using Sequences for tables.
	 5. Writing Procedures in PL/SQL Block (IN, OUT, INOUT, DEFAULT keywords).
	     a. Create an empty procedure, replace a procedure and call procedure
	     b. Create a stored procedure and call it
	     c. Define procedure to insert data
	     d. A forward declaration of procedure
	 6. Writing Functions in PL/SQL Block.
	     a. Define and call a function
	     b. Define and use function in select clause,
		1

	c. Call function in dbms_output.put_line
	  d. Recursive function
	7. Writing PL/SQL Blocks for Triggers.
	       a. Insert/Update/Delete Trigger
	       b. Before/After Trigger
	       c. Working with statement Level Trigger and Row Level
	  Trigger.
	       d. Remove Trigger
	8. Writing PL/SQL Block for Cursors
	       a. Cursor
	  attributes:%ROWCOUNT,%FOUND,%NOTFOUND,%ISOPEN
	        b. Cursor with sub queries
	        c. Combination of PL/SQL, cursor and for loop
	        d. Parameterized cursors, Cursor Variables
	9. Writing Exception Handling with PL/SQL.
	        a. Exception Types (implicitly raised, Explicitly raised)
	        b. Trapping Exceptions (WHEN exception1, WHEN OTHERS)
	        c. Predefined Exception
	        – NO_DATA_FOUND
	        – TOO_MANY_ROWS
	        – INVALID_CURSOR
	        – ZERO_DIVIDE
	        – DUP_VAL_ON_INDEX
	10. Indexes: Creating, dropping, and maintaining indexes on tables for
	the given column.	

References:
1. Ivan Bayross, "SQL,PL/SQL -The Programming language of Oracle", B.P.B. Publications ,
    a. 4th edition.
2. Michael Abbey, Michael J. Corey, Ian Abramson, Oracle 8i – A Beginner's Guide, TataMcGraw-Hill, 3rd edition
3. PL/SQL Language Reference 11g, , Sheila Moore, E. Belden, 2nd edition.
4. Ramakrishnam, Gehrke, "Database Management Systems", McGraw- Hill, 3rd edition.
Additional References:
● Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education
● Robert Sheldon, Geoff Moes, Beginning MySQL, Wrox Press.
● Joel Murach, Murach's MySQL, Murach
● https://docs.oracle.com

Course Code: RUSCS305

Course Title: COMBINATORICS AND GRAPH THEORY

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION
	After Completing this course student will be able to :

CO 1	Appraise the beauty of combinatorics and how combinatorial problems naturally arise in many settings.
CO 2	Associate the combinatorial features in real world situations and Computer Science applications.
CO 3	Apply combinatorial and graph theoretical concepts to understand Computer Science concepts and apply them to solve problems.

DETAILED SYLLABUS

Course Code	Unit	COMBINATORICS AND GRAPH THEORY	Lectures
RUSCS305	**I**	**Introduction to Combinatorics:** Definition, Combinatorics and Graph Theory/ Number Theory/Geometry and Optimization, **Strings, Sets, and Binomial Coefficients:** Strings, Permutation & Combination, The Binomial coefficients, Binomial Theorem, Multinomial Coefficients. **Induction:** Mathematical Induction, and Inductive Definitions Proofs by Induction., Pigeonhole principle, Inclusion and Exclusion Principle	15 L
	II	**Graph theory:** Basic Notation and Terminology ,Matrix representation of Graph, Types of graphs, Degree of vertex, Eulerian Trails and Circuits, Hamiltonian Paths and Cycles, Applications of graph theory, Connectivity, Coverings, Isomorphism. **Trees and Forest**: Spanning Tree and Kirchoff's Theorem. **Planar graphs:** Planar graphs, Counting labeled trees, Euler's formula, Kuratowski's Theorem. **Graph Coloring:** Coloring and its example, chromatic number, chromatic polynomial, Bipartite graphs, Matching in Bipartite Graphs ,Ramsey Number Theory.	15 L
	III	**Network Flow:** Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm ,Maximum Flow in a Transport Network: The Ford–Fulkerson Algorithm **Graph Algorithms**: Reachability: Warshall's Algorithm , Depth-First and Breadth-First Searches ,The Lightest Path: Dijkstra's Algorithm , Floyd's Algorithm , The Lightest Spanning Tree: Kruskal's and Prim's Algorithms , The Lightest Hamiltonian Circuit (Travelling Salesman's Problem): The Annealing Algorithm and the Karp–Held Heuristics , Maximum Matching in Bipartite Graphs: The Hungarian Algorithm	15 L

Course Code	COMBINATORICS AND GRAPH THEORY	Credits
RUSCSP305	Following Practicals can be implemented using R/Python etc. 1. Solving problems on strings, sets and binomial coefficients. 2. Solving problems using induction. 3. Solving problems on Eulerian and Hamiltonian graphs.	**1**

4.	Solving problems on Chromatic number and coloring
5.	Solving problems using Kruskal's Algorithm
6.	Solving problems using Prim's Algorithm
7.	Solving problems using Dijkstra's Algorithm
8.	Solving problems of finding augmenting paths in network flows.
9.	Solving problems on network flows using Ford-Fulkerson Labeling Algorithm
10.	Solving problems on Reachability: Warshall's Algorithm , Depth-First and Breadth-First Searches.

References:
1. Applied Combinatorics, Mitchel T. Keller and William T. Trotter, 2016, http://www.rellek.net/appcomb.

Additional References:
- Applied Combinatorics, 6th edition, Alan Tucker, Wiley; (2016)
- Graph Theory and Combinatorics, Ralph P. Grimaldi, Pearson Education; Fifth edition (2012)
- Combinatorics and Graph Theory, John Harris, Jeffry L. Hirst, Springer(2010).
- Graph Theory: Modeling, Applications and Algorithms, Agnarsson, Pearson Education India (2008).

Course Code: RUSCS306

Course Title: PHYSICAL COMPUTING AND IOT PROGRAMMING

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION After Completing this course student will be able to :
CO 1	Define IoT and identify the applications of IoT.
CO 2	Describe the architecture of IoT.
CO 3	Identify various components required to build an IoT product.
CO 4	Design and assemble various components of Raspberry Pi Kit.
CO 5	Develop an IoT product.

DETAILED SYLLABUS

Course Code	Unit	PHYSICAL COMPUTING AND IOT PROGRAMMING	Lectures
RUSCS306	**I**	**Introduction to Networks:** N/w Types& Topologies, Protocols (TCP/IP), Attacks. **Introduction to IoT:** What is IoT? IoT examples, Simple IoT LED Program. **IoT and Protocols** **IoT Security:** HTTP, UPnp, CoAP, MQTT, XMPP**.** **IoT Service as a Platform:** Clayster, Thinger.io, SenseIoT, carriots and Node RED.	15 L

		IoT Security and Interoperability: Risks, Modes of Attacks, Tools for Security and Interoperability.	
	II	**Programming Raspberry Pi** **Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands **Programming interfaces:** Introduction to Node.js, Python. **Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.	15 L
	III	**SoC and Raspberry Pi** **System on Chip:** What is System on chip? Structure of System on Chip. **SoC products:** FPGA, GPU, APU, Compute Units. **ARM 8 Architecture:** SoC on ARM 8. ARM 8 Architecture Introduction **Introduction to Raspberry Pi:** Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. **Raspberry Pi Boot:** Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.	15 L

Course Code	PRACTICAL OF PHYSICAL COMPUTING AND IOT PROGRAMMING	Credits
RUSCSP306	1. Preparing Raspberry Pi: Hardware preparation and Installation 2. Linux Commands: Exploring the Raspbian 3. GPIO: Light the LED with Python 4. GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas 5. SPI: Camera Connection and capturing Images using SPI 6. Real Time Clock display using PWM. 7. Stepper Motor Control: PWM to manage stepper motor speed. 8. Node RED: Connect LED to Internet of Things 9. Stack of Raspberry Pi for better Computing and analysis 10. Create a simple Web server using Raspberry Pi	**1**

References:
1. Learning Internet of Things, Peter Waher, Packt Publishing(2015)
2. Internet of Things Principles and Paradigms, Rajkumar Buyya & Amir Vahid Dastjerdi, MK.
3. Mastering the Raspberry Pi, Warren Gay, Apress(2014)
4. Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013.

Additional References:
● Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly

Course Code: RUSCS307

Course Title: SKILL ENHANCEMENT: WEB PROGRAMMING

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Design valid, well-formed, scalable, and meaningful pages using emerging technologies.
CO 2	Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites
CO 3	Develop and implement client-side and server-side scripting language programs.
CO 4	Develop and implement Database Driven Websites.
CO 5	Design and apply XML to create a markup language for data and document centric applications.

DETAILED SYLLABUS

Course Code	Unit	SKILL ENHANCEMENT: WEB PROGRAMMING	Lectures
RUSCS307	**I**	**HTML & HTML5:**Fundamental Elements of HTML, Formatting Text in HTML, Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMS in HTML, Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page **CSS:** Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element	15 L
	II	**JavaScript:** Using JavaScript in an HTML Document, Programming Fundamentals of JavaScript – Variables, Operators, Control Flow Statements, Popup Boxes, Functions – Defining and Invoking a Function, Defining Function arguments, Defining a Return Statement, Calling Functions with Timer, JavaScript Objects - String, RegExp, Math, Date, Browser Objects - Window, Navigator, History, Location, Document, Cookies, Document Object Model, Form Validation using JavaScript **XML:** Comparing XML with HTML, Advantages and Disadvantages of XML, Structure of an XML Document, XML Entity References, DTD, XSLT.	15 L
	III	**AJAX:**AJAX Web Application Model, How AJAX Works, XMLHttpRequest Object – Properties and Methods, Handling	15 L

	asynchronous requests using AJAX
	PHP: Variables and Operators, Program Flow, Arrays, Working with Files and Directories, Working with Databases, Working with Cookies, Sessions and Headers
	Introduction to jQuery: Fundamentals, Selectors, methods to access HTML attributes, methods for traversing, manipulators, events, effects.

Course Code	**PRACTICAL OF SKILL ENHANCEMENT: WEB PROGRAMMING**	**Credits**

RUSCSP307	1. Design a webpage that makes use of
	   a. Document Structure Tags
	   b. Various Text Formatting Tags
	   c. List Tags
	   d. Image and Image Maps
	2. Design a webpage that makes use of
	   a. Table tags
	   b. Form Tags (forms with various form elements)
	   c. Navigation across multiple pages
	   d. Embedded Multimedia elements
	3. Design a webpage that make use of Cascading Style Sheets with
	   a. CSS properties to change the background of a Page
	   b. CSS properties to change Fonts and Text Styles
	   c. CSS properties for positioning an element
	4. Write JavaScript code for
	   a. Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number
	   b. Validating the various Form Elements
	5. Write JavaScript code for
	   a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date
	   b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document, Storing and Retrieving Cookies
	6. Create a XML file with Internal / External DTD and display it using
	   a. CSS
	   b. XSL
	7. Design a webpage to handle asynchronous requests using AJAX on
	   a. Mouseover
	   b. Button click
	8. Write PHP scripts for
	   a. Retrieving data from HTML forms
	   b. Working with Arrays
	   c. Working with Files (Reading / Writing)
	9. Write PHP scripts for Working with Databases
	   a. Storing Records and Display them
	10. Design a webpage with some jQuery animation effects
		1

References:
1. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press
2. Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India(2017).
3. PHP: A Beginners Guide, Vikram Vaswani, TMH, 1ᶳᵗ edition

Additional References:
- HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY
- Learn to Master HTML 5, scriptDemics, StarEdu Solutions Pvt Ltd.
- Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly
- PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley

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 student will upload his practicals in the form of documents along with the screen shots of output on an 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 III

Course	301, 302, 303, 304, 305, 306, 307		
	Internal	**External**	**Total**
Theory	40	60	**700 (7 Papers)**
Practicals	20	30	**300 (6 Papers)**
Individual Semester Total			**1000**

Course Code: RUSCS401

Course Title: FUNDAMENTALS OF ALGORITHMS

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION After Completing this course student will be able to :
CO 1	Analyse the basic principles of algorithm design and why algorithm analysis is important
CO 2	Transform new problems into algorithmic problems with efficient solutions
CO 3	Compare different algorithm design techniques for solving problems
CO 4	Develop algorithms and implement them in Python

DETAILED SYLLABUS

Course Code	Unit	FUNDAMENTALS OF ALGORITHMS	Lectures
RUSCS401	**I**	**Introduction:** Introduction to algorithm, Why to analysis algorithm, Running time analysis, How to Compare Algorithms, Rate of Growth, Commonly Used Rates of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-Ω Notation, Theta-Θ Notation, Asymptotic Analysis, Properties of Notations, Commonly used Logarithms and Summations, Performance characteristics of algorithms, Master Theorem for Divide and Conquer, Master Theorem: Problems & Solutions. **Recursion and Backtracking:** Introduction, What is Recursion, Why Recursion, Format or a Recursive Function, Recursion and Memory (Visualization), Recursion versus Iteration, Notes on Recursion, Example Algorithms of Recursion, What is Backtracking? Example Algorithms of Backtracking **Tree algorithms:** What is a Tree? Glossary, Binary Trees, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals, Generic Trees (N-ary Trees), Threaded Binary Tree Traversals, Expression Trees, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees .	15 L
	II	**Graph Algorithms:** Introduction, Glossary, Applications of Graphs, Graph Representation, Graph Traversals, Topological Sort, Shortest Path Algorithms, Minimal Spanning Tree **Selection Algorithms:** What are Selection Algorithms? Selection by Sorting, Partition-based Selection Algorithm, Linear Selection Algorithm - Median of Medians Algorithm, Finding the K Smallest Elements in Sorted Order.	15 L

		Algorithms Design Techniques: Introduction, Classification, Classification by Implementation Method, Classification by Design Method	
	III	**Greedy Algorithms:** Introduction, Greedy Strategy, Elements of Greedy Algorithms, Advantages and Disadvantages of Greedy Method, Greedy Applications, Understanding Greedy Technique **Divide and Conquer Algorithms:** Introduction, What is Divide and Conquer Strategy? Divide and Conquer Visualization, Understanding Divide and Conquer, Advantages of Divide and Conquer, Disadvantages of Divide and Conquer, Master Theorem, Divide and Conquer Applications **Dynamic Programming:** Introduction, What is Dynamic Programming Strategy? Properties of Dynamic Programming Strategy, Problems which can be solved using Dynamic Programming, Dynamic Programming Approaches, Examples of Dynamic Programming Algorithms, Understanding Dynamic Programming, Longest Common Subsequence . **Complexity Classes:** Introduction, Polynomial/Exponential Time, What is a Decision Problem?, Decision Procedure, What is a Complexity Class?, Types of Complexity Classes, Reductions	15 L

Course Code	FUNDAMENTALS OF ALGORITHMS	**Credits**
RUSCSP401	1. Write a Python program to perform matrix multiplication. Discuss the complexity of the algorithm used. 2. Write a Python program to sort n names using Quick sort algorithm. Discuss the complexity of the algorithm used. 3. Write a Python program to sort n numbers using Merge sort algorithm. Discuss the complexity of the algorithm used. 4. Write a Python program for inserting an element into a binary tree. 5. Write a Python program for deleting an element (assuming data is given) from a binary tree. 6. Write Python program for checking whether a given graph G has simple path from source s to destination d. Assume the graph G is represented using adjacent matrix. 7. Write a Python program for finding the smallest and largest elements in an array A of size n using the Selection algorithm. Discuss Time complexity. 8. Write a Python program for finding the second largest element in an array A of size n using Tournament Method. Discuss Time complexity. 9. Write a Python program for implementing Huffman Coding Algorithm. Discuss the complexity of the algorithm. 10. Write a Python program for implementing Strassen's Matrix multiplication using Divide and Conquer method. Discuss the complexity of algorithm	**1**

References:
1. Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi , CareerMonk Publications, 2016
2. Introduction to Algorithm, Thomas H Cormen, PHI,3ʳᵈ edition

Additional References:
- Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 2016, Wiley
- Fundamentals of Computer Algorithms, Sartaj Sahni and Sanguthevar Rajasekaran Ellis Horowitz, Universities Press

Course Code: RUSCS402

Course Title: ADVANCED JAVA

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION
	After Completing this course student will be able to :
CO 1	Develop and Explore advanced topics of Java programming for solving problems.
CO 2	Understand the concepts related to Java Technology
CO 3	Explore and understand use of Java Server Programming

DETAILED SYLLABUS

Course Code	Unit	ADVANCED JAVA	Lectures
RUSCS402	**I**	**Event Handling:** The Delegation Event Model, Event classes (ActionEvent, FocusEvent, InputEvent, ItemEvent, KeyEvent, MouseEvent, MouseWheelEvent, TextEvent, and WindowEvent) and various listener interfaces (ActionListener, FocusListener, ItemListener, KeyListener, MouseListener, MouseMotionListener, MouseWheelListener, TextListener, WindowFocusListener, WindowListener). **JDBC:** Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updations, CallableStatement, BLOB & CLOB	15 L
	II	**Servlets:** Introduction, Web application Architecture, Http Protocol & Http Methods, Web Server & Web Container, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, ServletConfig, ServletContext, Servlet Communication, Session Tracking Mechanisms **JSP:** Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions:	15 L

	III	**JSON:** Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java **Hibernate:** Introduction, Writing the application, application development approach, creating database and tables in MySQL, creating a web application, Adding the required library files, creating a java bean class, creating hibernate configuration and mapping file, adding a mapping resource, creating JSPs. **Struts 2:** Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results & Result Types, Value Stack/OGNL	15 L

Course Code	PRACTICAL OF ADVANCED JAVA	Credits
RUSCSP402	1. Practical on event handling using swing components. 2. Practical on JDBC Component. 3. Develop a Java application to store image in a database as well as retrieve image from database. 4. Write a Java application to demonstrate servlet life cycle. 5. Design database for student administration. Develop servlet(s) to perform CRUD operations. 6. Write a Java application to demonstrate JSP applications. 7. Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write a Java application to access JavaBeans Properties. 8. Design application using Struts2. Application must accept the user name and greet the user when the command button is pressed. 9. Write a Java application to encoding and decoding JSON in Java. 10. Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database.	**1**

References:
1. Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR,9th Edition
2. Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill,5th Edition
3. Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) ,3rd Edition

Additional References:
- Advanced Java Programming, Uttam K. Roy, Oxford University Press
- *The Java Tutorials: http://docs.oracle.com/javase/tutorial/)*
- The Java Tutorials of Sun Microsystems Inc

Course Code: RUSCS403

Course Title: COMPUTER NETWORKS

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION
	After Completing this course student will be able to :
CO 1	Enumerate the layers functionality of the TCP/IP Model
CO 2	Apply Protocols to Network design and implementation for specific use case
CO 3	Evaluate and assess the use of network components, protocols in real networks.
CO 4	Use tools and technologies for network design, monitoring and maintenance.

DETAILED SYLLABUS

Course Code	Unit	COMPUTER NETWORKS	Lectures
RUSCS403	I	**Introduction Network Models:** Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet, standards and administration Internet Standards. Network Models, Protocol layering, Scenarios, Principles of Protocol Layering, Logical Connections, TCP/IP Protocol Suite, Layered Architecture, Layers in the TCP/IP Protocol Suite, Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing. Detailed introduction to Physical Layer, Detailed introduction to Data-Link Layer, Detailed introduction to Network Layer, Detailed introduction to Transport Layer, Detailed introduction to Application Layer. Data and Signals, Analog and Digital Data, Analog and Digital Signals, Sine Wave Phase, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length, Transmission of Digital Signals, Transmission Impairments, Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay)	15 L
	II	**Introduction to Physical Layer and Data-Link Layer:** Digital Transmission digital-to-digital conversion, Line Coding, Line Coding Schemes, analog-to-digital conversion, Pulse Code Modulation (PCM), Transmission Modes, Parallel Transmission, Serial Transmission. Analog Transmission, digital-to-analog Conversion, Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, analog-to-analog Conversion, Amplitude Modulation (AM), Frequency Modulation (FM), Phase	15 L

		Modulation (PM), Multiplexing, Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Time-Division Multiplexing. Transmission Media, Guided Media, Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable. Switching, Three Methods of Switching , Circuit Switched Networks, Packet Switching, Introduction to Data-Link Layer, Nodes and Links, Services, Two Sub-layers, Three Types of addresses, Address Resolution Protocol (ARP). Error Detection and Correction, introduction, Types of Errors, Redundancy, Detection versus Correction.	
	III	**Network layer, Transport Layer** Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA. Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer Switches, Routers, Introduction to Network Layer, network layer services, Packetizing, Routing and Forwarding, Other Services, IPv4 addresses, Address Space, Classful Addressing. Unicast Routing, General Idea, Least-Cost Routing, Routing Algorithms, Distance-Vector Routing, Link-State Routing, Path-Vector Routing, Introduction to Transport Layer, Transport-Layer Services, Connectionless and Connection-Oriented Protocols. Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.	15 L

Course Code	PRACTICAL OF COMPUTER NETWORKS	**Credits**
RUSCSP403	1. Understanding the working of NIC cards, Ethernet/Fast Ethernet/Gigabit Ethernet. 2. Crimping of Twisted-Pair Cable with RJ45connector for Straight-Through, Cross-Over, Roll-Over. 3. To understand their respective role in networks/internet. 4. Problem solving with IPv4, which will include the concept of Classful addressing. (supportive Hint: use Cisco Binary Game) 5. Using linux-terminal or Windows-cmd, execute following networking commands and note the output: *ping, traceroute, netstat, arp, ipconfig*. 6. Create a basic network of two computers using appropriate network wire. 7. Connect multiple (min.6) computers using layer 2 switch. 8. Connect a network in triangular shape with three layer two switches and every switch will have four computers. Verify their connectivity with each other. 9. Create a wireless network of multiple PCs using appropriate access points. 10. Using Wireshark, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working.	**1**

References:
1. Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013.
2. Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson Education, 2011.

Additional References:
- Computer Network, Bhushan Trivedi, Oxford University Press
- Data and Computer Communication, William Stallings, PHI

Course Code: RUSCS404

Course Title: SOFTWARE ENGINEERING
Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Understanding the disciplinary process to develop software and to know different software testing methods.
CO 2	Illustrate the different phases in software development.
CO 3	Interpret project management and risk management process.
CO 4	Shows how to apply software testing methods.

DETAILED SYLLABUS

Course Code	Unit	SOFTWARE ENGINEERING	Lectures
RUSCS404	**I**	**Software Engineering Fundamentals**: Introduction to Software Engineering, Types of Software, System Development Approaches. **Software Development Life Cycle Models:** SDLC, Prescriptive Process Model, Specialized Process Model. **Changing trends in software development:** Unified process & its phases, Agile Development, Extreme Programming and SCRUM. **The Analyst as Project Manager**: Project Management, Project Management Knowledge Areas, Project Initiation & Project Planning, Project Scheduling, Project Feasibility Study, Staffing & Launching the Project. **Software Requirements Specification:** Introduction to SRS, Components of SRS, Characteristics of SRS, Investigating System Requirements.	15 L
	II	**System Analysis:** Events & event table, Introduction to UML, Class Diagram, Use Case Diagram - Use Case Scenario, Interaction Diagram, Activity Diagram, State-chart Diagram. **System Design:** Design Class Diagram, Package Diagram,	15 L

		Component Diagram, Deployment Diagram. **Project Management Process:** Software Configuration Management Process, Change Management Process, CMM, Risk Management, RMMM Plan. **Software Measurement and Metrics:** Product Metrics, Function-Based Metrics, Operation-Oriented Metrics, Halstead Metrics Applied to Testing, Empirical Estimation Models – COCOMO II, Estimation for Agile Development.	
	III	**Software Quality Assurance:** Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, The ISO 9000 Quality Standards. **Software Testing Fundamentals:** Purpose, Goals & Objective of Testing, Challenges & Issues in Testing, Types of Testing, Software Testing Terminologies. **Black Box Testing:** Introduction, Equivalence partitioning, Boundary value analysis, Robustness testing, Cause Effect Graph. **White Box Testing:** Statement Coverage, Branch/Decision Coverage, Condition Coverage, Graph Matrix, Cyclomatic complexity, Mutation Testing. **Planning Software Testing:** Test Plan, Test Plan Specification, Test Case Execution and Analysis, Defect logging and tracking.	15 L

References:
1. System Analysis and Design in the Changing World, Satzinger, Jackson, Burd, Thomas Learning
2. System Analysis and Design in the Changing World, Satzinger, Jackson, Burd, CengageLearning (India Edition)
3. Software Engineering, A Practitioner's Approach, Roger S, Pressman.(2014).
4. An Integrated Approach to Software Engineering, Pankaj Jalote, Narosa,3rd edition.
5. Software Testing - Concepts & Practices, K. Mustafa, R. A. Khan, Narosa,Reprint 2009

Course Code: RUSCS405

Course Title: LINEAR ALGEBRA

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Appraise the relevance of linear algebra in the field of computer science.
CO 2	Explain the concepts and solve them through program implementation
CO 3	Evaluate various matrix operations and interpret their results.

DETAILED SYLLABUS

Course Code	Unit	LINEAR ALGEBRA	Lectures
RUSCS405	**I**	**Field:** Introduction to complex numbers, numbers in Python , Abstracting over fields, Playing with GF(2) **Vector:** Vectors are functions, Vector addition, Scalar-vector multiplication, Combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product. Solving $Ax = 0$ and $Ax = b$. **Vector Space**: Linear combination of vectors, Span, The geometry of sets of vectors, Vector spaces and subspaces, Linear Dependence and Independence.	15 L
	II	**Matrix:** Matrices as vectors, Transpose, Matrix-vector and vector-matrix multiplication in terms of linear combinations, Matrix-vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrix-matrix multiplication, Inner product and outer product, From function inverse to matrix inverse **Basis:** Coordinate systems, Two greedy algorithms for finding a set of generators, Minimum Spanning Forest and GF(2), Basis, Unique representation, Change of basis **Dimension:** Dimension and rank, Direct sum, Dimension and linear functions.	15 L
	III	**Gaussian elimination:** Row Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination. **Inner Product:** The inner product for vectors over the reals, Orthogonality, **Orthogonalization:** Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, Orthogonal complement. **Eigenvector:** Modeling discrete dynamic processes, Diagonalization of the Fibonacci matrix, Eigenvalues and eigenvectors, Coordinate representation in terms of eigenvectors, The Internet worm, Markov chains, Modeling a web surfer: PageRank.	15 L

Course Code	PRACTICAL OF LINEAR ALGEBRA	Credits

RUSCSP405	1. Write a program which demonstrates the following:
	    a. Addition of two complex numbers
	    b. Displaying the conjugate of a complex number
	    c. Plotting a set of complex numbers
	    d. 0, 180, 270 degrees and also by scaling by a number a=1/2, a=1/3, a=2 etc. Creating a new plot by rotating the given number by a degree 9
	2. Write a program to do the following:
	    a. Enter a vector u as a n-list
	    b. Enter another vector v as a n-list
	    c. Find the vector au+bv for different values of a and b
	    d. Find the dot product of u and v
		1

3. Write a program to do the following:
   a.   Enter two distinct faces as vectors u and v.
   b.   Find a new face as a linear combination of u and v i.e. au+bv
   for a and b in R.
   c.   Find the average face of the original faces.
4. Write a program to do the following:
   a.   Enter an r by c matrix M (r and c being positive integers)
   b.   Display M in matrix format
   c.   Display the rows and columns of the matrix M
   d.   Find the scalar multiplication of M for a given scalar.
   e.   Find the transpose of the matrix M.
5. Write a program to do the following:
   a.   Find the vector –matrix multiplication of a r by c matrix M
   with a c-vector u.
   b.   Find the matrix-matrix product of M with a c by p matrix N.
6. Write a program to enter a matrix and check if it is invertible.
          If the inverse exists, find the inverse.
7. Write a program to convert a matrix into its row echelon
          form.
8. Write a program to do the following:
   a.   Enter a positive number N and find numbers a and b such that
   $a^2 - b^2 = N$
    b.   Find the gcd of two numbers using Euclid's algorithm.
9. Write a program to do the following:
   a.   Enter a vector b and find the projection of b orthogonal to a
   given vector u.
   b.   Find the projection of b orthogonal to a set of given vectors
10. Write a program to enter a given matrix and an eigenvalue of the
same. Find its eigenvector.

References:
1. Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013)
Additional References:
●   Linear Algebra and Probability for Computer Science Applications, Ernest Davis, A K Peters/CRC Press (2012).
●   Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition (2007).
●   Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition (2002).

Course Code: RUSCS406

Course Title: .NET Technologies
Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION
	After Completing this course student will be able to :
CO 1	Identify and compare various .Net Frameworks and its features.

CO 2	Develop proficiency in C# programming language.
CO 3	Choose various controls for designing a web page.
CO 4	Integrating Front-end with Back-end SQL Server.
CO 5	Develop fully functional websites using .NET Technologies.

DETAILED SYLLABUS

Course Code	Unit	.NET Technologies	Lectures
RUSCS406	I	**Introduction to .NET:** Introduction to .Net Framework, .NET advantages, .Net Various Framework Components - CLR, CTS, MSIL, Class Library, JIT Compiler, Memory Management, Garbage Collection and its phases. **Introduction to C#:** Comments, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods, Classes, Value Types and Reference Types, Namespaces and Assemblies, Inheritance, Static Members, Casting Objects, Partial Classes **ASP.NET:** Introduction to ASP.NET, Introduction to web applications, ASP.NET Architecture - ASP.Net application Life Cycle - Application Life Cycle & Page Life Cycle, Anatomy of a Web Form - Page Directive, Doctype, Writing Code - Code-Behind Class, Adding Event Handlers, Anatomy of an ASP.NET Application - ASP.NET File Types, ASP.NET Web Folders **ASP.NET Server Controls**- HTML Server Controls, Web Server Controls, Page Class, Global.asax, Web.config	15 L
	II	**ASP.NET Controls:** Standard Controls, Validation Controls, Navigation Controls, Login Controls, Events & Properties of Various Controls - AutoPostBack **Rich Controls:** Calendar Control, AdRotator Control, MultiView Control **Themes and Master Pages:** How Themes Work, Applying a Simple Theme, Handling Theme Conflicts, Simple Master Page and Content Page, Connecting Master pages and Content Pages, Master Page with Multiple Content Regions, Master Pages and Relative Paths **ASP.NET AJAX:** ScriptManager, Partial Refreshes, Progress Notification, Timed Refreshes **State Management:** ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State **Caching:** When to Use Caching, Output Caching, Data Caching	15 L
	III	**Working With Data:** ADO.NET Fundamentals, Data Binding, The Data Controls, **Working with Files & Streams:** Files & Streams, **Working with XML:** XML Classes – XMLTextWriter, XMLTextReader **LINQ:** Understanding LINQ, LINQ Basics **Introduction to MVC Framework :** MVC Architecture & its	15 L

		Features, MVC Components, MVC Application Folders, Working with Controls	

Course Code	PRACTICAL OF .NET Technologies	Credits

RUSCSP406	1. Write C# programs for understanding C# basics involving
	    a. Variables and Data Types b. Object-Based Manipulation
	    b. Conditional Logic d. Loops e. Methods
	2. Write C# programs for Object oriented concepts of C# such as:
	    a. Program using classes b. Constructor and Function Overloading
	    b. Inheritance d. Namespaces
	3. Design ASP.NET Pages with
	    a. Server controls.
	    b. Web controls and demonstrate the use of AutoPostBack
	    c. Rich Controls (Calendar / Ad Rotator)
	4. Design ASP.NET Pages for State Management using
	    a. Cookies
	    b. Session State
	    c. Application State
	5. Perform the following activities
	    a. Design ASP.NET page and perform validation using various Validation Controls
	    b. Design an APS.NET master web page and use it other (at least 2-3) content pages.
	    c. Design ASP.NET Pages with various Navigation Controls
	6. Performing ADO.NET data access in ASP.NET for
	    a. Simple Data Binding
	    b. Repeated Value Data Binding
	7. Design ASP.NET application for Interacting (Reading / Writing) with XML documents
	8. Design ASP.NET Pages for Performance improvement using Caching
	9. Design ASP.NET application to query a Database using LINQ
	10. Design and use AJAX based ASP.NET pages.
		1

References:
1. Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012)
Additional References:
- The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill
- Beginning ASP.NET 4 in C# and VB Imar Spanajaars, WROX

Course Code: RUSCS407

Course Title: SKILL ENHANCEMENT: ANDROID DEVELOPER FUNDAMENTALS
Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Understand the requirements of Mobile programming environment.
CO 2	Learn about basic methods, tools and techniques for developing Apps
CO 3	Explore and practice App development on Android Platform
CO 4	Develop working prototypes of working systems for various uses in daily lives.
CO 5	Developing applications running on smart mobile devices

DETAILED SYLLABUS

Course Code	Unit	SKILL ENHANCEMENT: ANDROID DEVELOPER FUNDAMENTALS	Lectures
RUSCS407	I	What is Android? Obtaining the required tools, creating first android app, understanding the components of screen, adapting display orientation, action bar, Activities and Intents, Activity Lifecycle and Saving State, Basic Views: TextView, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views, ProgressBar View, AutoCompleteTextView, TimePicker View, DatePicker View, ListView View, Spinner View	15 L
	II	User Input Controls, Menus, Screen Navigation, RecyclerView, Drawables, Themes and Styles, Material design, Providing resources for adaptive layouts, AsyncTask and AsyncTaskLoader, Connecting to the Internet, Broadcast receivers, Services, Notifications, Alarm managers, Transferring data efficiently	15 L
	III	Data - saving, retrieving, and loading: Overview to storing data, Shared preferences, SQLite primer, store data using SQLite database, ContentProviders, loaders to load and display data, Permissions, performance and security, Firebase and AdMob, Publish your app	15 L

Course Code	PRACTICAL OF SKILL ENHANCEMENT: ANDROID DEVELOPER FUNDAMENTALS	Credits
RUSCSP407	1. Install Android Studio and Run Hello World Program. 2. Create an android app with Interactive User Interface using Layouts. 3. Create an android app that demonstrates working with TextView Elements. 4. Create an android app that demonstrates Activity Lifecycle and Instance State. 5. Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers. 6. Create an android app that demonstrates the use of an Options Menu. 7. Create an android app that demonstrates Screen Navigation Using the App Bar and Tabs. 8. Create an android app to Connect to the Internet and use BroadcastReceiver. 9. Create an android app to show Notifications and Alarm manager. 10. Create an android app to save user data in a database and use of different queries.	**1**

References:
    1. "Beginning Android 4 Application Development", Wei-Meng Lee, March 2012, WROX.
Additional References:
- https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-concepts/content/en/Unit%202/41_c_user_input_controls.html

- https://developers.google.com/training/courses/android-fundamentals
- https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-practicals/details

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 **2Hrs** 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).

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): <u>60% - 30 Marks</u>
 <u>30 Marks Practical Questions:</u>
  ● Students have to acquire at least 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 IV

Course	401, 402, 403, 404, 405, 406, 407		
	Internal	**External**	**Total**
Theory	**40**	**60**	**700 (7 Papers)**
Practicals	**20**	**30**	**300 (6 Papers)**
Individual Semester Total		**1000**	

Course Code: RUSCS501

Course Title: ARTIFICIAL INTELLIGENCE
Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Define Artificial Intelligence.
CO 2	Implement various algorithms to solve the real world problems.
CO 3	Use various tools and techniques for automation.
CO 4	Identify different areas for AI implementation.
CO 5	Produce own algorithms.

DETAILED SYLLABUS

Course Code	Unit	ARTIFICIAL INTELLIGENCE	Lectures
RUSCS501	I	**What Is AI**: Foundations, History and State of the Art of AI. Intelligent Agents: Agents and Environments, Nature of Environments, Structure of Agents. Problem Solving by searching: Problem-Solving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.	15 L
	II	**Learning from Examples**: Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and Choosing the Best Hypothesis, Theory of Learning, Regression and Classification with Linear Models, Artificial Neural Networks, Nonparametric Models, Support Vector Machines, Ensemble Learning, Practical Machine Learning	15 L
	III	**Learning probabilistic models**: Statistical Learning, Learning with Complete Data, Learning with Hidden Variables: The EM Algorithm. Reinforcement learning: Passive Reinforcement Learning, Active Reinforcement Learning, Generalization in Reinforcement Learning, Policy Search, Applications of Reinforcement Learning.	15 L

Course Code	PRACTICAL OF ARTIFICIAL INTELLIGENCE	Credits
RUSCSP501	1. Implement Breadth first search algorithm for Romanian map	**1**

	problem.
	2. Implement Iterative deep depth first search for Romanian map problem.
	3. Implement A* search algorithm for Romanian map problem.
	4. Implement recursive best-first search algorithm for Romanian map problem.
	5. Implement a decision tree learning algorithm for the restaurant waiting problem.
	6. Implement feed forward back propagation neural network learning algorithm for the restaurant waiting problem.
	7. Implement Adaboost ensemble learning algorithm for the restaurant waiting problem.
	8. Implement Naive Bayes' learning algorithm for the restaurant waiting problem.
	9. Implement passive reinforcement learning algorithm based on adaptive dynamic programming (ADP) for the 3 by 4 world problem
	10. Implement passive reinforcement learning algorithm based on temporal differences (TD) for 3 by 4 world problems.

Reference:
1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig,3rd Edition, Pearson, 2010.

Additional Reference(s):
1) Artificial Intelligence: Foundations of Computational Agents, David L Poole,Alan K. Mackworth, 2nd Edition, Cambridge University Press ,2017.
2) Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017
3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013

Course Code: RUSCS502

Course Title: SOFTWARE TESTING AND QUALITY ASSURANCE

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION After Completing this course student will be able to :
CO 1	Testing Methods can be used as an effective tools in providing quality assurance concerning for software
CO 2	Skills to design test case plan for testing software

CO 3	Understand various software testing methods and strategies
CO 4	Understand a variety of software metrics, and identify defects and managing those defects
CO 5	Design SQA activities, SQA strategy, formal technical review report for software Quality control and assurance.

DETAILED SYLLABUS

Course Code	Unit	SOFTWARE TESTING AND QUALITY ASSURANCE	Lectures
RUSCS502	I	**Software Testing and Introduction to quality:** Introduction, Nature of errors, an example for Testing, Definition of Quality , QA, QC, QM and SQA , Software Development Life Cycle, Software Quality Factors Software Testing Techniques: Testing Fundamentals, Test Case Design, White Box Testing and its types, Black Box Testing and its types. **Software Testing Strategies:** Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing	15 L
	II	**Software Metrics:** Concept and Developing Metrics, Different types of Metrics, Complexity metrics. **Verification and Validation:** Definition of V&V , Different types of V & V Mechanisms, Concepts of Software Reviews, Inspection and Walkthrough Defect Management: Definition of Defects, Defect Management Process, Defect Reporting, Metrics Related to Defects, Using Defects for Process Improvement.	15 L
	III	**Test Techniques:** Equivalence Partitioning, Boundary Value Analysis, Decision Tables, State-Based Testing and State Transition Diagrams, State Transition Tables, Control-Flow Testing, Statement Coverage, Decision Coverage, Loop Coverage, Path Testing, Cyclomatic Complexity, Data Flow Testing Structure-Based Testing **Quality Improvement:** Introduction, Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts **Quality Costs :** Defining Quality Costs, Types of Quality Costs, Quality Cost Measurement, Utilizing Quality Costs for Decision-Making	15 L

Course Code	PRACTICAL OF SOFTWARE TESTING AND QUALITY ASSURANCE	Credits
RUSCSP502	1. Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.	**1**

2. Conduct a test suite for any two web sites.

3. Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP.

4. Write and test a program to login a specific web page.

5. Write and test a program to update 10 student records into table into Excel file

6. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).

7. Write and test a program to provide the total number of objects present / available on the page.

8. Write and test a program to get the number of items in a list / combo box.

9. Write and test a program to count the number of checkboxes on the page checked and unchecked count.

10. Load Testing using JMeter, Android Application testing using Appium Tools, Bugzilla Bug tracking tools.

References:

1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4th Edition,, Pearson Education, 2005

2. Software Engineering - A Practitioner's Approach, Roger S. Pressman, 5th Edition, Tata McGraw Hill, 2001

3. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall, 2010.

4. Total Quality Management, Dale H. Besterfield, 3rd Edition, Prentice Hall, 2003.

5. Advanced Software Testing—Vol. 3 by Rex Black and Jamie L. Mitchell, Rocky Nook Publication

Additional Reference(s):

- Software engineering: An Engineering approach, J.F. Peters, W. Pedrycz , John
- Wiley,2004
- Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik,
- Priyadarshi Tripathy , John Wiley & Sons, Inc. , Publication, 2008
- Software Engineering and Testing, B. B. Agarwal, S. P. Tayal, M. Gupta, Jones and
- Bartlett Publishers, 2010

Course Code: RUSCS503

Course Title: INFORMATION AND NETWORK SECURITY

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Analyze Particular information and network security Problems

CO 2	Identify generic security threats and Vulnerabilities
CO 3	Assess and Apply various cryptographic techniques
CO 4	Enumerate security Protocols at Network ,Transport and Application Layers of TCP/IP model
CO 5	Implement security solutions for confidentiality, Authentication and privacy.

DETAILED SYLLABUS

Course Code	Unit	INFORMATION AND NETWORK SECURITY	Lectures
RUSCS503	I	**Introduction to Network Security:** Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms **Classical Encryption Techniques:** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers **Public-Key Cryptography and RSA:** Principles of Public-Key Cryptosystems, The RSA Algorithm	15 L
	II	**Key Management:** Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange **Message Authentication and Hash Functions:** Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC Digital Signatures and Authentication: Digital Signatures, Authentication Protocols, Digital Signature Standard, Digital Envelope. **Authentication Applications:** Kerberos, X.509 Authentication, Public-Key Infrastructure.	15 L
	III	**Electronic Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management Web Security: Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction **Intrusion:** Intruders, Intrusion Techniques, Intrusion Detection **Malicious Software:** Viruses and Related Threats, Virus Countermeasures, DDOS **Firewalls:** Firewall Design Principles, Types of Firewalls	15 L

Course Code	PRACTICAL OF INFORMATION AND NETWORK SECURITY	Credits
RUSCSP503	1. Write programs to implement the following Substitution Cipher Techniques: ● Caesar Cipher ● Monoalphabetic Cipher 2 Write programs to implement the following Substitution Cipher Techniques: ● Vernam Cipher ● Playfair Cipher 3 Write programs to implement the following Transposition Cipher Techniques: ● Rail Fence Cipher ● Simple Columnar Technique 4 Write program to encrypt and decrypt strings using ● DES Algorithm ● AES Algorithm 5 Write a program to implement an RSA algorithm to perform encryption / decryption of a given string. 6 Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys. 7 Write a program to implement the MD5 algorithm to compute the message digest. 8 Write a program to calculate HMAC-SHA1 Signature 9 Write a program to implement SSL. 10 Configure Windows Firewall to block: ● A port ● An Program ● A website	**1**

References:
1) Cryptography and Network Security: Principles and Practice 5th Edition, William Stallings, Pearson,2010

Additional Reference(s):
● Cryptography and Network Security, Atul Kahate, Tata McGraw-Hill, 2013.
● Cryptography and Network, Behrouz A Fourouzan, Debdeep Mukhopadhyay, 2nd Edition,TMH,2011

Course Code: RUSCS504

Course Title: WEB SERVICES

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Understand the details of web services technologies like SOAP, WSDL, and UDDI
CO 2	Identify how to implement and deploy web service clients and servers.
CO 3	Understand the design principles and application of SOAP and REST based web services (JAX-Ws and JAX-RS) Services
CO 4	Evaluate Spring web services and develop spring applications.
CO 5	Understand WCF service to develop secure web services and QoS of Web Services

DETAILED SYLLABUS

Course Code	Unit	WEB SERVICES	Lectures
RUSCS504	**I**	**Web services basics :** What Are Web Services? Types of Web Services Distributed computing infrastructure, , Building Web Services with JAX-WS, Registering and Discovering Web Services, Service Oriented Architecture, Web Services Development Life Cycle	15 L
	II	**Spring Web Services:** Spring WS – Overview, Spring WS - Static WSDL, Spring WS – Writing Server, Spring WS - Unit Test Server, Spring WS - Writing Client, Spring WS - Unit Test Client, use of web Service in Android and testing	15 L
	III	**Developing Service-Oriented Applications with WCF :** What Is Windows Communication Foundation, Fundamental Windows Communication Foundation Concepts, Windows Communication Foundation Architecture, WCF and .NET Framework Client Profile, Basic WC Programming, WCF Feature Details. Web Service QoS	15 L

Course Code	PRACTICAL OF WEB SERVICES	Credits
RUSCSP504	1. Write a program to implement to create a simple web service that converts the temperature from Fahrenheit to Celsius and	**1**

| | vice versa.
2. Write a program to implement the operation can receive request and will return a response in two ways.
    a) One - Way operation
    b) Request -Response
3. Write a program to implement business UDDI Registry entry.
4. Develop client which consumes web services developed in different platforms.
5. Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.
6. Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.
7. Define a RESTful web service that accepts the details to be stored in a database and performs CRUD operation.
8. Implement a typical service and a typical client using WCF.
9. Use WCF to create a basic ASP.NET Asynchronous JavaScript and XML (AJAX) service.
10. Demonstrates using the binding attribute of an endpoint element in WCF.
11. Demonstrate practicals on Spring in Web services | |

References:

1) Web Services: Principles and Technology, Michael P. Papazoglou, Pearson Education Limited, 2008
2) RESTful Java Web Services, Jobinesh Purushothaman, PACKT Publishing,2ⁿᵈ Edition, 2015
3) DevelopingService-Oriented Applications with WCF, Microsoft, 2017 https://docs.microsoft.com/en-us/dotnet/framework/wcf/index

Additional Reference(s):

- Leonard Richardson and Sam Ruby, RESTful Web Services, O'Reilly, 2007
- The Java EE 6Tutorial, Oracle, 2013.

Course Code: RUSCS505

Course Title: SKILL ENHANCEMENT: ETHICAL HACKING

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION
	After Completing this course student will be able to :
CO 1	Explain ethics, methodologies of ethical hacking.

CO 2	Identify security vulnerabilities and weaknesses in the target applications.
CO 3	Test and exploit systems using various tools
CO 4	Illustrate different types of attacks and its countermeasure

DETAILED SYLLABUS

Course Code	Unit	SKILL ENHANCEMENT: ETHICAL HACKING	Lectures
RUSCS505	I	**Information Security**: Asset, Threat, Vulnerability, Attack, Exploit, Types of malware, Access Control, CIA, Risk. **Introduction to Ethical Hacking:** Objective of Ethical, Hacking, Asset, Vulnerability Threat, Expolit Ethical hacking types, Hacker types, Vulnerability assessment and Penetration Testing. **Phases of Ethical hacking**: Footprinting, Reconnaince, scanning, Enumeration.	15 L
	II	**Types of vulnerabilities**: OWASP Top 10 : cross-site scripting (XSS), cross site request forgery (CSRF/XSRF), SQL injection, input parameter manipulation, broken authentication, sensitive information disclosure, XML External Entities, Broken access control, Security Misconfiguration, Using components with known vulnerabilities, Insufficient Logging and monitoring, OWASP Mobile Top 10, CVE Database **Vulnerability Assessment and Penetration Testing (VAPT) Process:** Introduction to VA and PT, Threat modeling, Categories of Penetration Test, Tools used like WebInspect/Qualys, Nessus, Differences in VA and PT.	15 L
	III	**Types of attacks and their common prevention mechanisms** : Keystroke Logging, Denial of Service (DoS /DDoS), Waterhole attack, brute force, phishing and fake WAP, Eavesdropping, Man-in-the-middle, Session Hijacking,, Cookie Theft, URL Obfuscation, buffer overflow, DNS poisoning, ARP poisoning, Identity Theft, IOT Attacks, BOTs and BOTNETs, Keylogging, Buffer Overflows, Privilege Escalation, ARP Poisoning, Password Cracking, WEP Vulnerabilities, MAC Spoofing, MAC Flooding, IPSpoofing, SYN Flooding, Smurf attack, Applications hacking : SMTP/Email-based attacks, VOIP vulnerabilities, Directory traversal, Input Manipulation, , SQL injection, XSS, Intellectual property theft, Vulnerability Assessment and Penetration Testing (VAPT) Process,Instant messenger threats,Evading IDS	15 L

		Enforcement of security: Firewall, Secure coding practices, Security policy	

Course Code	PRACTICAL OF SKILL ENHANCEMENT: ETHICAL HACKING	Credits/ Lectures

RUSCSP505	1. Use Google and Whois for Reconnaissance
	2. a) Use CrypTool to encrypt and decrypt passwords using RC4 algorithm
	   b) Use Cain and Abel for cracking Windows account password using Dictionary attack and to decode wireless network passwords
	3. a) Run and analyze the output of following commands in Linux - ifconfig, ping, netstat,traceroute
	   b) Perform ARP Poisoning in Windows
	4. Use NMap scanner to perform port scanning of various forms - ACK, SYN, FIN, NULL, XMAS
	5. Use Wireshark (Sniffer) to capture network traffic and analyze
	6. Simulate persistent cross-site scripting attack
	7. Session impersonation using Firefox and Tamper Data add-on
	8. Perform SQL injection attack
	9. Create a simple logger using python
		1

References

1) Certified Ethical Hacker Study Guide v9, Sean-Philip Oriyano, Sybex; Study Guide Edition,2016
2) CEH official Certified Ethical Hacking Review Guide, Wiley India Edition, 2007

Additional Reference(s):
- http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines
- https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project
- https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10
- https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents
- https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_ Guide
- https://cve.mitre.org/
- https://access.redhat.com/blogs/766093/posts/2914051
- http://resources.infosecinstitute.com/applications-threat-modeling/#gref
- http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html

Course Code: RUSCSP506

Course Title: PRACTICAL OF ADVANCED WEB PROGRAMMING

Academic year 2020-21

Course Code	PRACTICAL OF ADVANCED WEB PROGRAMMING	Credits
RUSCSP506	1. Write a program to read the data & display it on the page simultaneously. 2. Write a program to change the name displayed on the textbox. 3. Write a program using ng-bind. 4. Working with filters. 5. Exploring AngularJS services. 6. Program using AngularJS tables. 7. Working with AngularJS Events. 8. Working with AngularJS forms & validations. 9. Exploring AngularJS Animations 10. Develop an application using AngularJS	**1**

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 **2Hrs** 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 student will upload his practicals in the form of documents along with the screen shots of output on an 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 V

Course	501, 502, 503, 504, 505, 506		
	Internal	**External**	**Total**
Theory	40	60	500 (5 Papers)
Practicals	20	30	300 (6 Papers)
Individual Semester Total			800

Course Code: RUSCS601

Course Title: CLOUD COMPUTING

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Define and describe cloud computing and its architecture.
CO 2	Compare various cloud services available in the cloud.
CO 3	Choose the appropriate cloud services for various types of applications.
CO 4	Explain various of cloud computing such as security, privacy, and interoperability.
CO 5	Express the concepts, key technologies strengths and limitations of cloud computing.

DETAILED SYLLABUS

Course Code	Unit	CLOUD COMPUTING	Lectures
RUSCS601	**I**	**Introduction to Cloud Computing:** Characteristics and benefits of Cloud Computing, Basic concepts of Distributed Systems, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing. Elements of Parallel Computing. Elements of Distributed Computing. Technologies for Distributed Computing. Cloud Computing Architecture. The cloud reference model. Infrastructure as a service. Platform as a service. Software as a service. Types of clouds.	15 L
	II	**Virtualization:** Characteristics of Virtualized Environments. Taxonomy of Virtualization techniques. Virtualization and Cloud Computing. Pros and Cons ofVirtualization. Virtualization using KVM, Creating virtual machines, oVirt - management tool for virtualization environment. Open challenges of Cloud Computing.	15 L
	III	**Introduction to OpenStack technologies:** OpenStack test-drive, Basic OpenStack operations, OpenStack CLI and APIs, Tenant model operations, Quotas, Private cloud building blocks, Controller deployment, Networking deployment, Block Storage deployment, Compute deployment, deploying and utilizing OpenStack in production environments, Building a production environment, Application orchestration using OpenStack Heat.	15 L

Course Code	PRACTICAL OF CLOUD COMPUTING	Credits
RUSCSP601	1. Implementation of Infrastructure as a Service	**1**

	a. VMWare Esxi Server b. Citrix Xenserver 2. Implementation of Remote Procedure Calls 3. Implementation of Remote Method Invocation on Local machine as well as Remote machine 4. Implementing Hadoop & Map Reduce 5. Application Development using Google App Engine 6. Installation and configuration of virtualization using KVM 7. Installation and configuration of OpenStack 8. Study of AWS, Google Cloud & Windows Azure	

References:
1. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013
1. OpenStack in Action, V. K. CODY BUMGARDNER, Manning Publications Co, 2016

Additional Reference(s):
- OpenStack Essentials, Dan Radez, PACKT Publishing, 2015
- OpenStack Operations Guide, Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, O'Reilly Media, Inc., 2014
- https://www.openstack.org

Course Code: RUSCS602

Course Title: CYBER FORENSIC

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Explain the procedures for identification, preservation, and extraction of electronic evidence
CO 2	Illustrate procedure of network and mobile forensics
CO 3	Analyze legal aspect and Ethics in digital forensics
CO 4	Evaluate real time case study in digital forensics

DETAILED SYLLABUS

Course Code	Unit	CYBER FORENSIC	Lectures
RUSCS602	**I**	**Computer Forensics :** Introduction to Computer Forensics, Understanding Computer Investigations, Data Acquisition. Processing Crime and Incident Scenes.	15 L

		Network Forensic : Introduction to Network Forensics and its challenges, Types of Digital evidence, Sources of Network based Evidence, Sources of log. **Cell Phone and Mobile Device Forensics:** Overview, Acquisition Procedures for Cell Phones and Mobile Devices.	
	II	**Internet Forensic :** Introduction to Internet Forensics, World Wide Web Threats, Hacking and Illegal access, Obscene and Indecent transmission, Domain Name Ownership Investigation, Reconstructing past internet activities and events **E-mail Forensics :** e-mail analysis, e-mail headers and spoofing, Laws against e-mail Crime, Messenger Forensics **Social Media Forensics:** Social Media Investigations **Browser Forensics:** Cookie Storage and Analysis, Analyzing Cache and temporary internet files, Web browsing activity reconstruction	15 L
	III	**Legal aspects and Ethics of Digital Forensics:** Expert Testimony in High-Tech Investigations, Information Technology Act. **Case Study :** Cyber Crime cases	15 L

Course Code	PRACTICAL OF CYBER FORENSIC	Credits
RUSCSP602	**Make use of Forensics tools to perform following:** 1.Evidence acquisition 2. Cyber Forensics Case examination 3. Network Forensics 4. Network Tracking and Process Monitoring 5. Mobile Forensics 6. Email Forensics 7. Browser Forensics 8. Write a program for Database backup and its restoration.	**1**

References:

1. Guide to computer forensics and investigations, Bill Nelson, Amelia Philips and Christopher Steuart, course technology,5th Edition,2015
2. Network Forensics, Sherri Davidoff, Jonathan HAM, Prentice Hall, 2012.

Additional Reference(s):

- Introduction to Social Media Investigation A Hands-on Approach, Jennifer Golbeck Judith L. Klavans, Technical Editor

Course Code: RUSCS603

Course Title: INFORMATION RETRIEVAL

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Understanding of information retrieval and its relationship to search engines
CO 2	Apply information retrieval models on live dataset.
CO 3	Able to handle issues in classical and web information retrieval
CO 4	Implementation of systems for gathering, indexing, and searching documents and of methods for evaluating systems.

DETAILED SYLLABUS

Course Code	Unit	INFORMATION RETRIEVAL	Lectures
RUSCS603	**I**	**Introduction to Information Retrieval:** Introduction, History of IR, Components of IR, and Issues related to IR, Boolean retrieval, Dictionaries and tolerant retrieval.	15 L
	II	**Link Analysis and Specialized Search:** Link Analysis, hubs and authorities, Pagerank and HITS algorithms, Similarity, Hadoop & MapReduce, Evaluation, Personalized search, Collaborative filtering and content-based recommendation of documents and products, handling "invisible" Web, Snippet generation, Summarization, Question Answering, Cross-Lingual Retrieval.	15 L
	III	**Web Search Engine:** Web search overview, web structure, the user, paid placement, search engine optimization/spam, Web size measurement, search engine optimization/spam, Web Search Architectures. **XML retrieval:** Basic XML concepts, Challenges in XML retrieval, A vector space model for XML retrieval, Evaluation of XML retrieval, Text-centric versus data-centric XML retrieval.	15 L

Course Code	PRACTICAL ON INFORMATION RETRIEVAL	Credits
RUSCSP603	**Practical may be done using software/tools like Python / Java / Hadoop** 1. Write a program to demonstrate bitwise operation. 2. Implement Page Rank Algorithm.	**1**

3. Implement Dynamic programming algorithm for computing the edit distance between strings s1 and s2. (Hint. Levenshtein Distance)
4. Write a program to Compute Similarity between two text documents.
5. Write a map-reduce program to count the number of occurrences of each alphabetic Character in the given dataset. The count for each letter should be case-insensitive (i.e., include both upper-case and lower-case versions of the letter; Ignore non-alphabetic Characters).
6. Implement a basic IR system using Lucene.
7. Write a program for Pre-processing of a Text Document: stop word removal.
8. Write a program for mining Twitter to identify tweets for a specific period and identify trends and named entities.
9. Write a program to implement a simple web crawler.
10. Write a program to parse XML text, generate Web graph and compute topic specific page rank.

References:

1) Introduction to Information Retrieval, C. Manning, P. Raghavan, and H. Schütze, Cambridge University Press, 2008
2) Modern Information Retrieval: The Concepts and Technology behind Search, Ricardo Baeza -Yates and Berthier Ribeiro - Neto, 2nd Edition, ACM Press Books 2011.
3) Search Engines: Information Retrieval in Practice, Bruce Croft, Donald Metzler and Trevor Strohman, 1st Edition, Pearson, 2009.

Additional Reference(s):

- Information Retrieval Implementing and Evaluating Search Engines, Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack, The MIT Press; Reprint edition (February 12, 2016)

Course Code: RUSCS604

Course Title: DATA SCIENCE

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION **After Completing this course student will be able to :**
CO 1	Develop Analytical solution to Real world data science Problem
CO 2	Apply statistical and mathematical techniques to explore data
CO 3	Compare and Implement various supervised and unsupervised Learning algorithms for specific use cases.
CO 4	Evaluate results of Analytics and suggest Solutions
CO 5	Articulate techniques for handling Time series and Semi-structured data

DETAILED SYLLABUS

Course Code	Unit	DATA SCIENCE	Lectures
RUSCS604	**I**	**Introduction to Data Science:** What is Data? Different kinds of data, Data Science Process or lifecycle. **Data Preprocessing:** Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation **Exploratory Data Analysis (EDA):** Measures of central tendency and dispersion, Bar plot, histogram, Box plots, stem-leaf diagram, multi-dimensional modeling	15 L
	II	**Statistical Modeling and Machine Learning Algorithms:** Introduction to model selection: Regularization, bias/variance tradeoff e.g.parsimony, AIC, BIC, Cross validation **Supervised Learning:** Regression, linear models, Regression trees, Time-series Analysis, Forecasting, Classification: classification trees, Logistic regression, separating hyperplanes, k-NN **Unsupervised Learning:** Principal Components Analysis (PCA), k-means clustering, Hierarchical clustering, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Clustering High-Dimensional Data, Constraint-Based Cluster Analysis, Outlier Analysis, Association rules from frequent itemsets. Ensemble methods: Increasing the Accuracy, Model Selection.	15 L
	III	**Semi-structured systems:** Semi-structured data Model, management and querying of data. **Unstructured data analytics systems:** Unstructured data model, NoSQL databases, Text Analytics **Big data Analytics:** What is Big data? ,Document shingling	15 L

Course Code	PRACTICAL OF DATA SCIENCE	Credits
RUSCSP604	1. Data Cleaning 2. Exploratory Data Analysis 3. Regression 4. Decision Tree 5. Principal Component Analysis 6. Clustering 7. Association 8. Model validation 9. NoSQL database 10. Document shingling	**1**

Reference(s):
1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013
2. J. Han and M. Kamber, " Data Mining: Concepts and Techniques", Second Edition, Elsevier, Reprinted2008
3. Elmasri and Navathe, "Fundamentals of Database Systems", Pearson Education
4. Hadoop The Definitive Guide, Tom White, O'Reilly

5. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, EMC Education Services
6. Big Data Analytics with R and Hadoop, Vignesh Prajapati, PACKT Publishing.
7. Mining of Massive Datasets, Anand Rajaraman and Jeffrey David Ullman, Cambridge University Press, 2012

Additional Reference(s):
- Hands-On Programming with R, Garrett Grolemund,1ˢᵗ Edition, 2014
- An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015

Course Code: RUSCS605

Course Title: OPTIMIZATION TECHNIQUES

Academic year 2020-21

COURSE OUTCOMES:

COURSE OUTCOME	DESCRIPTION
	After Completing this course student will be able to :
CO 1	Appraise Optimization Techniques on real world problems.
CO 2	Develop optimize solution to given problem
CO 3	Collect information to Implement optimal models for solving real world problems.

DETAILED SYLLABUS

Course Code	Unit	OPTIMIZATION TECHNIQUES	Lectures
RUSCS605	**I**	**Introduction to Operation Research:** Operation Research approach, scientific methods, introduction to models and modeling techniques, general methods for Operation Research models, methodology and advantages of Operation Research, history of Operation Research. **Linear Programming (LP):** Introduction to LP and formulation of Linear Programming problems, Graphical solution method, alternative or multiple optimal solutions, Unbounded solutions, Infeasible solutions, Maximization – Simplex Algorithm, Minimization – Simplex Algorithm using Big-M method, Two phase method, Duality in linear programming	15 L
	II	**Transportation & Assignment Problems:** Introduction to Transportation problems, various methods of Transportation problem, Variations in Transportation problem, introduction to Assignment problems, variations in Assignment problems. traveling salesman problem. **Integer LP Models**	15 L

		Gomary's Cutting plane algorithms, branch and bound technique for integer programming	
	III	**Sequencing:** Introduction, processing N jobs through two machines, processing N jobs through three machines, processing N jobs through m machines. **Theory of Games**: Introduction, Two person Zero sum Games, Games with Saddle point	15 L

References:
1. Operation research theory and Applications,J.K.Sharma,5th Edition, MacMillan Publishing Co
Additional Reference(s):
- Taha H.A. - Operations Research; An Introduction, 7th ed.,2003, MacMillan Publishing Co

Course Code: RUSCSP605
Course Title: PROJECT IMPLEMENTATION

Academic year 2020-21

Course Code	PROJECT IMPLEMENTATION	Credits

RUSCSP605	**Project Implementation Guidelines**
	1. A learner is expected to carry out one project: in Semester VI.
	2. A learner can choose any topic which is covered in Semester I- semester VI or any other Topic with the prior approval from the head of the department/ project in charge.
	3. The Project has to be performed individually.
	4. A learner is expected to devote a minimum 180hrs of efforts in the project.
	5. The project can be application oriented/web-based/database/research based.
	6. It has to be an implemented work; just theoretical study will not be acceptable.
	7. A learner can choose any programming language, computational techniques and tools Which have been covered during a BSc course or any other with the prior permission of the head of the department/ project guide.
	8. A project guide should be assigned to a learner. He/she will assign a schedule for the
	Project and hand it over to a learner. The guide should oversee the project progress on a weekly basis
	9. The quality of the project will be evaluated based on the novelty of the topic, scope of the work, relevance to computer science, adoption of emerging techniques/technologies and it's real-world application.
	10. A learner has to maintain a project report with the following subsections
	   a) Title Page
	   b) Certificate
	    A certificate should contain the following information –
	    -The fact that the student has successfully completed the project as per the syllabus
	     and that it forms a part of the requirements for completing the BSc degree in
		2

| | | computer science of University of Mumbai.
- The name of the student and the project guide
- The academic year in which the project is done
- Date of submission,
- Signature of the project guide and the head of the department with date along with the department stamp, Space for signature of the university examiner and date on which the project is evaluated.

c) Self-attested copy of the Plagiarism Report from any open source tool.

d) Index Page detailing description of the following with their subsections:

- Title: A suitable title giving the idea about what work is proposed.
- Introduction: An introduction to the topic giving proper background of the topic.
- Requirement Specification: Specify Software/hardware/data requirements.
- System Design details : Methodology/Architecture/UML/DFD/Algorithms/protocols etc. used(whichever is applicable)
- System Implementation: Code implementation

- Results: Test Cases/Tables/Figures/Graphs/Screenshots/Reports etc.
- Conclusion and Future Scope: Specify the Final conclusion and future scope
-References: Books, web links, research articles, etc.
11. The size of the project report shall be around twenty to twenty five pages, excluding the code.
12. The Project report should be submitted in a spiral bound form
13. The Project should be certified by the concerned Project guide and Head of the department.
14. A learner has to make a presentation of a working project and which will be evaluated. | |

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>
30 Marks Practical Questions:
- Students have to acquire at least 40% marks in each paper individually.

Particulars	Practical
Laboratory work	30
Total	**30**

Project Exam total marks: 100 Marks

INTERNAL COMPONENT - 40 Marks
- Project Proposal - 10 Marks
- Analysis Phase - 10 Marks
- Design Phase - 10 Marks
- Implementation - 10 Marks

Marking Scheme
- o Each student has to follow the schedule for above mentioned phases as given by the Project Guide.
- o Marks will be allotted on the basis of the presentation made by the student at each stage of project development.
- o Students has to maintain regular phases completion chart and project documentation duly signed By internal guide

EXTERNAL COMPONENT - 60 Marks
- Project Quality - 20 Marks.
- Working of Project - 20 Marks.
- Student Presentation - 20 Marks.

PASSING CRITERIA 40%: Student has to acquire minimum of 40% marks each course (Theory/Practical/Project).

Overall Examination & Marks Distribution Pattern

Semester- VI

Course			
	Internal	**External**	**Total**
Theory	40	60	500 (5 Papers)
Practicals	20	30	200 (4 Papers)
Project	40	60	100
Individual Semester Total			800