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

S. P. Mandali's

Ramnarain Ruia Autonomous College

(Affiliated to University of Mumbai)

Syllabus for SYB.Sc.

Program: B.Sc (Computer Science)

Program Code: Computer Science (RUSCS)

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

PROGRAM OUTLINE

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 Management Systems	2	**RUSCSP304**	Practical of Database Management Systems	1
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

Course Code: RUSCS301

Course Title: THEORY OF COMPUTATION

Academic year 2021-22

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.	15 L

		Undecidability: The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable 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 2021-22

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 accept a student's resume. (Design form)	**1**

	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 2021-22

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 Models	15 L
	II	**Process Synchronization:** General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution,	15 L

		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	
	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 2021-22

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; explicit	15 L

		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, c. Call function in dbms_output.put_line	**1**

	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 2021-22

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 ,	15 L

		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	

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. 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.	**1**

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 2021-22

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. **IoT Security and Interoperability:** Risks, Modes of Attacks, Tools for Security and Interoperability.	15 L
	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 2021-22

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 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.	15 L

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, 1st 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 2021-22

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	15 L

		Algorithm - Median of Medians Algorithm, Finding the K Smallest Elements in Sorted Order. **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 2021-22

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	15 L

		JSP: Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions: Standard actions and customized actions.	
	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 2021-22

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 Modulation (PM), Multiplexing,	15 L

		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 2021-22

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, Component Diagram, Deployment Diagram. **Project Management Process:** Software Configuration Management Process, Change Management Process, CMM, Risk Management, RMMM Plan.	15 L

		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 2021-22

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 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.	**1**

	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 a2 – b2 = 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 2021-22

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 Features, MVC Components, MVC Application Folders, Working with Controls	15 L

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 2021-22

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.	**1**

	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.	

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): 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 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**

Practicals	**20**	**30**	**200 (4 Papers)**
Project	**40**	**60**	**100**
Individual Semester Total			**800**