Resolution No: AC/II(20-21).2.RUS6

S.P. Mandali's

RAMNARAIN RUIA AUTONOMOUS COLLEGE

RUIA COLLEGE

Explore • Experience • Excel

Syllabus for: S.Y.B.Sc.

Program: B.Sc.

Course Code: Computer Science (RUSCS)

(Choice Based Credit System (CBCS) with effect from academic year 2018-19)

CREDIT BASED SYSTEM AND GRADING SYSTEM
ACADEMIC YEAR 2018-2019

SEMESTER – III (THEORY)

COURSE CODE	COURSE TYPE	COURSE TITLE	CREDITS	LECTURES/WEEK
RUSCS301	Core Subject	Theory of Computation	2	3
RUSCS302	Core Subject	Core JAVA	2	3
RUSCS303	Core Subject	Operating System	2	3
RUSCS304	Core Subject	Database Management Systems	2	3
RUSCS305	Core Subject	Combinatorics and Graph Theory	2	3
RUSCS306	Core Subject	Physical Computing and IoT Programming	2	3
RUSCS307	Core Subject	Skill Enhancement: Web Programming	2	3

SEMESTER – III (PRACTICALS)

COURSE CODE	COURSE TITLE	CREDITS	PRACTICALS /WEEK
RUSCSP302	Practical of Core JAVA	1	3
RUSCSP303	Practical of Operating System	1	3
RUSCSP304	Practical of Database Management Systems	1	3
RUSCSP305	Practical of Combinatorics and Graph Theory	1	3
RUSCSP306	Practical of Physical Computing and IoT Programming	1	3
RUSCSP307	Practical of Skill Enhancement: Web Programming	1	3

SEMESTER – IV (THEORY)

COURSE CODE	COURSE TYPE	COURSE TITLE	CREDITS	LECTURES/WEEK
RUSCS401	Core Subject	Fundamentals of Algorithms	2	3
RUSCS402	Core Subject	Advanced JAVA	2	3
RUSCS403	Core Subject	Computer Networks	2	3
RUSCS404	Core Subject	Software Engineering	2	3
RUSCS405	Core Subject	Linear Algebra	2	3
RUSCS406	Core Subject	.NET Technologies	2	3
RUSCS407	Core Subject	Skill Enhancement: Android Developer Fundamentals	2	3

SEMESTER – IV (PRACTICALS)

COURSE CODE	COURSE TITLE	CREDITS	PRACTICALS /WEEK
RUSCSP401	Practical of Fundamentals of Algorithms	1	3
RUSCSP402	Practical of Advanced JAVA	1	3
RUSCSP403	Practical of Computer Networks	1	3
RUSCSP405	Practical of Linear Algebra	1	3
RUSCSP406	Practical of .NET Technologies	1	3
RUSCSP407	Practical of Skill Enhancement: Android Developer Fundamentals	1	3

SEMESTER III - THEORY

COURSE CODE	COURSE NAME	CREDITS	LECTURE/ WEEK
RUSCS301	**RUSCS301** **THEORY OF COMPUTATION**	**2**	**3**

Learning Objective:
To provide the comprehensive insight into theory of computation by understanding grammar,languages and other elements of modern language design. Also to develop capabilities to designand develop formulations for computing models and identify its applications in diverse areas

Learning Outcome:
Upon completion of this course the student should be able to:

- Understand Grammar and Languages
- Learn about Automata theory and its application in Language Design
- Learn about Turing Machines and Pushdown Automata
- Understand Linear Bound Automata and its applications

UNITS	COURSE CONTENTS	NO. OF LECTURES
I	**Automata Theory:** Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite StateMachines, DFA and NDFA equivalence, Mealy and Moore Machines,Minimizing Automata. **Formal Languges:** Defining Grammar, Derivations, Languges generated by Grammar, Comsky 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 and CFG.	15 L
III	**Linear Bound Automata:** The Linear Bound Automata Model, Linear BoundAutomata and Languages. **Turing Machines:** Turing Machine Definition, Representations, Acceptabilityby Turing Machines, Designing and Description of Turing Machines, Turing Machine	15 L

	Construction, Variants of Turing Machine. **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	COURSE NAME	CREDITS	LECTURE/ WEEK
RUSCS302	**RUSCS302 CORE JAVA**	**2**	**3**

Learning Objective:

The objective of this course is to teach the learner how to use Object Oriented paradigm to develop code and understand the concepts of Core Java and to cover-up with the pre-requisites of Core java.

Learning Outcome:

Upon completion of this course the student should be able to:

- Object oriented programming concepts using Java.
- Knowledge of input, its processing and getting suitable output.
- Understand, design, implement and evaluate classes and applets.
- Knowledge and implementation of AWT package.

UNITS	COURSE CONTENTS	NO. OF LECTURES
I	**The Java Language:** Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays.	15 L

	OOPS: Introduction, Class, Object, Static Keywords, Constructors, this Key Word, Inheritance, super Key Word, 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	
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

References:

- 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	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS303	**RUSCS303 OPERATING SYSTEM**	**2**	**3**

Learning Objective:

- To introduce various components of computer hardware and operating systems.
- To discuss the structure of operating system, its functions and algorithms.

Learning Outcome:
Students completing this course will be able to:

- Understanding the working of operating system, its structures and functioning
- Compare various algorithms used in operating systems.

UNITS	COURSE CONTENTS	NO. OF LECTURES

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, 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

References:

- 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	COURSE NAME	CREDITS	LECTURE/ WEEK
RUSCS304	**RUSCS304** **DATABASE MANAGEMENT SYSTEMS**	**2**	**3**

Learning Objective:

To develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage.

Learning Outcome:

- Upon completion of this course the student should be able to:

- Learn about using PL/SQL for data management
- Master concepts of stored procedure and triggers and its use.
- Understand concepts and implementations of Exception handling
- To learn and understand Database Programming Paradigms

UNITS	COURSE CONTENTS	NO. OF LECTURES
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, Datetime 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, IFTHEN-ELSIF 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 cursor, Cursor for loops, Cursor variables, parameterized cursors, nested cursors, FOR UPDATE Clause and WHERE CURRENT Clause	15 L
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

References:

- Ivan Bayross, “SQL,PL/SQL -The Programming language of Oracle”, B.P.B. Publications , 4th edition.
- Michael Abbey, Michael J. Corey, Ian Abramson, Oracle 8i – A Beginner’s Guide, TataMcGraw-Hill, 3rd edition

- PL/SQL Language Reference 11g, , Sheila Moore, E. Belden, 2nd edition.
- 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, Begning MySQL, Wrox Press.
- Joel Murach, Murach’s MySQL, Murach
- https://docs.oracle.com

COURSE CODE	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS305	**RUSCS305** **COMBINATORICS AND GRAPH THEORY**	**2**	**3**

Learning Objective:
To give the learner a broad exposure of combinatorial Mathematics through applications especially the Computer Science applications.

Learning Outcome:
Upon completion of this course the student should be able to:

- Appreciate beauty of combinatorics and how combinatorial problems naturally arise in many settings.
- Understand the combinatorial features in real world situations and Computer Science applications.
- Apply combinatorial and graph theoretical concepts to understand Computer Science concepts and apply them to solve problems.

UNITS	COURSE CONTENTS	NO. OF LECTURES
I	**Introduction to Combinatorics:**_Enumeration, Combinatorics and Graph Theory/ Number Theory/Geometry and Optimization, Sudoku Puzzles. Strings, Sets, and Binomial Coefficients: Strings- A First Look, Combinations, Combinatorial, The Ubiquitous Nature of Binomial Coefficients, The Binomial, Multinomial Coefficients. Induction: Introduction, The Positive Integers are Well Ordered, The Meaning of Statements, Binomial Coefficients Revisited, Solving Combinatorial Problems Recursively, Mathematical Induction, and Inductive Definitions Proofs by Induction. Strong Induction	15 L
II	**Graph Theory:** Basic Notation and Terminology, Multigraphs : Loops and Multiple Edges, Eulerian and Hamiltonian Graphs, Graph Coloring, Planar Counting, Labeled Trees, A Digression into Complexity Theory. Applying Probability to Combinatorics, Small Ramsey Numbers, Estimating Ramsey Numbers, Applying Probability to Ramsey Theory, Ramsey’s Theorem The Probabilistic Method	15 L

III	**Network Flows:**_Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm, A Concrete Example, Integer Solutions of Linear Programming Problems. Combinatorial Applications of Network Flows: Introduction, Matching in Bipartite Graphs, Chain partitioning, Pólya’s Enumeration Theorem: Coloring the Vertices of a Square.	15 L

References:

- 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	COURSE NAME	CREDITS	LECTURE/WEEK
RUSCS306	**RUSCS306** **PHYSICAL COMPUTING AND IOT PROGRAMMING**	**2**	**3**

Learning Objective:

To learn about SoC architectures; Learn how Raspberry Pi. Learn to program Raspberry Pi. Implementation of internet of Things and Protocols.

Learning Outcome:

Upon completion of this course the student should be able to:

- Enable learners to understand System On Chip Architectures.
- Introduction and preparing Raspberry Pi with hardware and installation.
- Learn physical interfaces and electronics of Raspberry Pi and program them using practical’s
- Learn how to make consumer grade IoT safe and secure with proper use of protocols.

UNITS	COURSE CONTENTS	NO. OF LECTURES
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.	15L
II	**Programming Raspberry Pi** **Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands	15 L

	Programming interfaces:_Introduction to Node.js, Python. **Raspberry Pi Interfaces:**_UART, GPIO, I2C, SPI **Useful Implementations:**Cross Compilation, Pulse Width Modulation, SPI for Camera.	
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

References:

- Learning Internet of Things, Peter Waher, Packt Publishing(2015)
- Mastering the Raspberry Pi, Warren Gay, Apress(2014)
- Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013.

Additional References:

- Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly

COURSE CODE	COURSE NAME	CREDITS	LECTURE/ WEEK
RUSCS307	**RUSCS307** **WEB PROGRAMMING**	**2**	**3**

Learning Objective:
To provide insight into emerging technologies to design and develop state of - the art web applications using client-side scripting, server-side scripting, and database connectivity.

Learning Outcome:
Upon completion of this course the student should be able to:

- To design valid, well-formed, scalable, and meaningful pages using emerging technologies.
- Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites
- To develop and implement client-side and server-side scripting language programs.
- To develop and implement Database Driven Websites.
- Design and apply XML to create a markup language for data and document centric applications.

UNITS	COURSE CONTENTS	NO. OF LECTURES
I	**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	15L

	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	
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

References:

- HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press
- Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India(2017).
- 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

SEMESTER III - PRACTICALS

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP302	**RUSCSP302** **PRACTICAL OF - CORE JAVA**		
	1. Accept integer values for a, b and c which are coefficients of quadratic equation. Find the solution of 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 package animals create interface Animal with suitable behaviors. Implement the interface Animal in the same package animals. 5. Demonstrate Java inheritance using extends keyword. 6. Demonstrate method overloading and method overriding in Java. 7. Demonstrate creating your own exception in Java. 8. Using various swing components design Java application to accept a student's resume. (Design form) 9. Write a Java List example and demonstrate methods of Java List interface. 10. Design simple calculator GUI application using swing components.

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP303	**RUSCSP303** **PRACTICAL OF - OPERATING SYSTEMS**		
	1. Advanced Filtering using awk, sed. 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. Setting up local area network a. LAN topologies b. LAN equipment c. Networking with TCP/IP d. Configuring TCP/IP e. Adding windows computer’s to user LAN f. IP address classes 4. Connecting to the internet and configuring samba a. Setting up linux as a proxy server b. Configuring mozilla or firefox to use as a proxy c. Setting Up Samba Server 5. Server setup and configuration a. Setting up NFS file server b. The Apache web server c. Setting up FTP server 6. Understanding COMPUTER SECURITY: Firewall and security configurations a. LINUX security checklist b. Securing linux with IP table firewalls		

c. Configuring an IP table firewall

d. Securing Linux features

7. Using gcc compiler (Programming using C).

a. 5 to 10 programs

8. Using javac compiler (Implementing Socket programs).

5 to 10 programs

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP304	**RUSCSP304** **PRACTICAL OF - DATABASE MANAGEMENT SYSTEMS**		

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...ELSIF...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. Uncontrained 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

d. Recursive function

7. Writing PL/SQL Blocks for Trigger.

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.

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP305	**RUSCSP305** **PRACTICAL OF - COMBINATORICS AND GRAPH THEORY**		
	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 posets and their associated networks.		

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP306	**RUSCSP306** **PRACTICAL OF - PHYSICAL COMPUTING AND IoT PROGRAMMING**		
	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

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP307	**RUSCSP307** **PRACTICAL OF - WEB PROGRAMMING**		
	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, c. 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. Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number c. Working with Arrays d. Working with Files (Reading / Writing)		

	9. Write PHP scripts for a. Working with Databases (Storing Records / Reprieving Records and Display them) b. Storing and Retrieving Cookies c. Storing and Retrieving Sessions 10. Design a webpage with some jQuery animation effects.

SEMESTER IV - THEORY

COURSE CODE	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS401	**RUSCS401** **FUNDAMENTALS OF ALGORITHMS**	**2**	**3**

Learning Objective:

- To understand basic principles of algorithm design and why algorithm analysis is important
- To understand how to implement algorithms in Python
- To understand how to transform new problems into algorithmic problems with efficient solutions
- To understand algorithm design techniques for solving different problems

Learning Outcome:

Upon completion of this course the student should be able to:

- Understand the concepts of algorithms for designing good program
- Implement algorithms using Python

UNITS	COURSE CONTENTS	NO. OF LECTURES
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, Divide and Conquer Master Theorem: Problems & Solutions, Master Theorem for Subtract and Conquer Recurrences.	15 L
II	**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 **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
III	**Algorithms Design Techniques:** Introduction, Classification, Classification by Implementation Method, Classification by Design Method **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	15 L

	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	

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	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS402	**RUSCS402** **ADVANCED JAVA**	**2**	**3**

Learning Objective:

Explore advanced topic of Java programming for solving problems.

Learning Outcome:

Upon completion of this course the student should be able to:

1) Understand the concepts related to Java Technology
2) Explore and understand use of Java Server Programming

UNITS	COURSE CONTENTS	NO. OF LECTURES
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: Standard actions and customized actions.	15 L
III	**JSON :**Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java	15 L

	Web Service: Defining Client Access with Interfaces: Remote Access, Local Access, Local Interfaces and Container-Managed Relationships, Deciding on Remote or Local Access, Web Service Clients, Method Parameters and Access. Building Web Services with JAX-WS: Setting the Port, Creating a Simple Web Service and Client with JAX-WS. **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	

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	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS403	**RUSCS403 COMPUTER NETWORKS**	**2**	**3**

Learning Objective:

To Build an understanding of the fundamental concepts of computer networking and introduce the student to advanced networking concepts.

Learning Outcome:

Upon completion of this course the student should be able to:

1. Learner will be able to enumerate the layers functionality of the TCP/IP model.
2. Learner will be familiar with the basic protocols of computer networks, and how they can be used to assist in network design and implementation
3. Learner will acquire knowledge that will help them in Advanced courses and certifications in computer networking.

UNITS	COURSE CONTENTS	NO. OF LECTURES
I	**Introduction Network Models:** Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area	15 L

	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)	
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, 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,	15 L
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

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	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS404	**RUSCS404** **SOFTWARE ENGINEERING**	**2**	**3**

Learning Objective:
To understand disciplinary process to develop software and to know different software resting methods.

Learning Outcome:
Upon completion of this course the student should be able to:

- Understand the different phases in software development.
- Understand project management and risk management process
- Able to apply software testing methods

UNITS	COURSE CONTENTS	NO. OF LECTURES
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. **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.	15 L
III	**Software Quality Assurance:** Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, The ISO 9000 Quality Standards.	15 L

	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.	
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	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS405	**RUSCS405** **LINEAR ALGEBRA**	**2**	**3**

Learning Objective:
To offer the learner the relevant linear algebra concepts through computer science applications

Learning Outcome:

1. Appreciate the relevance of linear algebra in the field of computer science.
2. Understand the concepts through program implementation
3. Instill a computational thinking while learning linear algebra.

UNITS	COURSE CONTENTS	NO. OF LECTURES
I	**Field:** Introduction to complex numbers, numbers in Python , Abstracting over fields, Playing with GF(2) **Vector Space:** Vectors are functions, Vector addition, Scalar-vector multiplication, Combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product, Solving a triangular system of linear equations. Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise	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), Linear dependence, Basis, Unique representation, Change of basis first look, Computational problems involving finding a basis **Dimension:** Dimension and rank, Direct sum, Dimension and linear functions, The annihilator.	15 L

III	**Gaussian elimination:** Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination, Finding a basis for the null space, Factoring integers. **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, Existence of eigenvalues, Markov chains, Modeling a web surfer: PageRank.	15 L

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	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS406	**RUSCS406** **.NET TECHNOLOGIES**	**2**	**3**

Learning Objective:
To explore .NET technologies for designing and developing dynamic, interactive and responsive web applications.

Learning Outcome:
Upon completion of this course the student should be able to:

- Understand the .NET framework
- Develop a proficiency in the C# programming language
- Proficiently develop ASP.NET web applications using C#
- Use ADO.NET for data persistence in a web application

UNITS	COURSE CONTENTS	NO. OF LECTURES
I	**The .NET Framework:**.NET Languages, Common Language Runtime, .NET Class Library **C# Language Basics:** 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	15 L

	ASP.NET: Creating Websites, 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, HTML Server Controls- View State, HTML Control Classes, HTML Control Events, HtmlControl Base Class, HtmlContainerControl Class, HtmlInputControl Class, Page Class, global.asax File, web.config File	
II	**Web Controls:** Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle **State Management:**ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State **Validation:**Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions **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 **Website Navigation:**Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control	15 L
III	**ADO.NET:**Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access **Data Binding:** Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource **Data Controls:**GridView, Details View, Form View **Working with XML:** XML Classes – XMLTextWriter, XMLTextReader **Caching:**When to Use Caching, Output Caching, Data Caching **LINQ:** Understanding LINQ, LINQ Basics, **ASP.NET AJAX:**ScriptManager, Partial Refreshes, Progress Notification, Timed Refreshes	15 L

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	COURSE NAME	CREDITS	LECTURE /WEEK
RUSCS407	**RUSCS407** **ANDROID DEVELOPER FUNDAMENTALS**	**2**	**3**

Learning Objective:

To provide the comprehensive insight into developing applications running on smart mobile devices and demonstrate programming skills for managing task on mobile. To provide systematic approach for studying definition, methods and its applications for Mobile-App development.

Learning Outcome:

Upon completion of this course the student should be able to:

- Understand the requirements of Mobile programming environment.
- Learn about basic methods, tools and techniques for developing Apps
- Explore and practice App development on Android Platform
- Develop working prototypes of working systems for various uses in daily lives.

UNITS	COURSE CONTENTS	NO. OF LECTURES
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

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

SEMESTER IV - PRACTICALS

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP401	RUSCSP401 PRACTICAL OF - FUNDAMENTALS OF ALGORITHMS		

	1. Write Python program to perform matrix multiplication. Discuss the complexity of algorithm used. 2. Write Python program to sort n names using Quick sort algorithm. Discuss the complexity of algorithm used. 3. Write Python program to sort n numbers using Merge sort algorithm. Discuss the complexity of algorithm used. 4. Write Python program for inserting an element into binary tree. 5. Write Python program for deleting an element (assuming data is given) from 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 Python program for finding the smallest and largest elements in an array A of size n using Selection algorithm. Discuss Time complexity. 8. Write Python program for finding the second largest element in an array A of size n using Tournament Method. Discuss Time complexity. 9. Write Python program for implementing Huffman Coding Algorithm. Discuss the complexity of algorithm. 10. Write Python program for implementing Strassen's Matrix multiplication using Divide and Conquer method. Discuss the complexity of algorithm.

COURSE CODE	**COURSE NAME**	**CREDITS**	**LECTURE/WEEK**
		1	**3**
RUSCSP402	**RUSCSP402** **PRACTICAL OF - ADVANCED JAVA**		
	1. Practical on event handling using swing component. 2. Practical on JDBC Component. 3. Develop 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 Java application to access JavaBeans Properties. 8. Design application using Struts2. Application must accept user name and greet user when command button is pressed. 9. Write Java application to encoding and decoding JSON in Java. 10. Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database.		

COURSE CODE	**COURSE NAME**	**CREDITS**	**LECTURE/WEEK**
		1	**3**
RUSCSP403	**RUSCSP403**		

	PRACTICAL OF - COMPUTER NETWORKS
	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 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 computer. Verify their connectivity with each other. 9. Create a wireless network of multiple PCs using appropriate access point. 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.

COURSE CODE	**COURSE NAME**	**CREDITS**	**LECTURE/WEEK**
		1	**3**
RUSCSP405	**RUSCSP405** **PRACTICAL OF - LINEAR ALGEBRA**		
	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. Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number a=1/2, a=1/3, a=2 etc. 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. 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 an 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 eigen value of the same. Find its eigen vector.

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP406	**RUSCSP406** **PRACTICAL OF - .NET TECHNOLOGIES**		
	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.		

COURSE CODE	COURSE NAME	CREDITS	LECTURE/WEEK
		1	3
RUSCSP407	**RUSCSP407** **PRACTICAL OF - ANDROID DEVELOPER FUNDAMENTALS**		
	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 demonstrate 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.		

MODALITY OF ASSESSMENT

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

B) External examination - 60 %

External Examination- 60 Marks Duration 2 Hrs

Theory Question Paper Pattern:-

		All Questions are Compulsory	
Questions	**Options**	**Based On**	**Marks**
Q1	*Any 3 out of 5*	Unit I, II, & III	15
Q2	*Any 3 out of 5*	Unit I	15
Q3	*Any 3 out of 5*	Unit II	15
Q4	*Any 3 out of 5*	Unit III	15

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

Practical Examination Pattern:

(A) Internal Examination: Internal Practical - 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

Heading	Practical I
Individual Practical Implementation & Performance	*10*

Design and implement innovative application of the technology	*10*
Total	20

(B) External (Semester end practical examination): <u>30 Marks</u>

<u>30 Marks Practical Questions:</u>

- Student has to acquire atleast 40% marks in each paper individually.

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

Particulars	**Practical 1**
Laboratory work	*30*
Total	**30**

PRACTICAL BOOK/JOURNAL

The students are required to present a duly certified journal for appearing at the practical examination, failing which they will not be allowed to appear for the examination.

In case of loss of Journal and/ or Report, a Lost Certificate should be obtained from Head/ Co-ordinator / Incharge of the department; failing which the student will not be allowed to appear for the practical examination.

Overall Examination and Marks DistributionPattern

Semester- III

Course	*RUSCS301,302,303,304,305,306,307.*			
	Internal	External	Total	
Theory	40	60	700	200
Practicals	20	30	300	100

Semester- IV

Course	*RUSCS401, 402,403,404,405,406,407.*			
	Internal	External	Total	
Theory	40	60	700	200
Practicals	20	30	300	100