SRI VENKATESWARA UNIVERSITY
DEGREE COURSE IN BACHELOR OF COMPUTER APPLICATIONS (BCA)

ARTIFICIAL INTELLIGNECE & DATA SCIENCE

New Course Introduced under CBCS W.E.F. 2021-22

SCHEME OF INSTRUCTIONS AND EXAMINATIONS
III - SEMESTER

S.No	Paper Code	Subject	Hours/ Week	No of Credits	Max.Marks Internal assessment	Max. Marks University Exam	Total Marks
1.		English – III	4	3	25	75	100
2.		Language(H/T/S) – III	4	3	25	75	100
3.		Life Skill Course – III	2	2	-0-	50	50
4.		Life Skill Course – IV	2	2	-0-	50	50
5		Skill Development Course – IV	2	2	-0-	50	50
6	C7	Operating Systems	4	3	25	75	100
7	C8	OOPS with Java	4	3	25	75	100
8	C8-P	Java Programming Lab	2	2	-0-	50	50
9	C9	R Programming	4	3	25	75	100
10	C9-P	R Programming Lab	2	2	-0-	50	50
Total			30	25	125	625	750

<h1 style="text-align:center">C7-OPERATING SYSTEM</h1>

OBJECTIVES:
- The operating system is the most important program that runs on a computer.
- Every general-purpose computer must have an operating system to run other programs.
- Operating systems perform basic tasks, such as recognizing input from the keyboard.
- Sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers.
- This course covers the concept of operating system and its applications.

OUTCOMES:
- After learning the fundamental concepts in Operating system including how OS has evolved over the years and different components of OS.
- Students will continue to more significant functions of OS like Process management, storage and memory management etc.
- This will provide the necessary information for students to extract maximum benefits out of the OS while developing programs, working with applications and etc.
- To enable the students to have a hands-on practical exposure to the Linux Red Hat Enterprise and make them prepared for the RHCE Certification.
- Student will be able to understand the security and protection of operating system. · Students can learn storage management

UNIT I. INTRODUCTION TO OPERATING SYSTEM

Objectives and Functions of OS, Evolution of OS in the pretext of evolution of UNIX, OS Structures, OS Components, OS Services, System calls, System programs, commands to monitor and perform the operations, Virtual Machines.

UNIT II PROCESS MANAGEMENT – PROCESSES AND THREADS

Processes in Unix environment: Process concept, Process scheduling, Co-operating processes, Inter process Communication Threads in Unix environment: Introduction to Threads, Single and Multi-threaded processes CPU Scheduling in Unix environment: Basic concepts, Scheduling criteria, Scheduling Algorithms, Multiple Processor Scheduling, Real-time Scheduling

UNIT III PROCESS MANAGEMENT – SYNCHRONIZATION AND DEADLOCKS

Process Synchronization in Unix: Mutual Exclusion, Critical – section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, Critical Regions, Monitors, OS Synchronization, Atomic Transactions. Deadlocks: System Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

UNIT IV STORAGE MANAGEMENT

Memory Management in the Unix environment: Logical and physical Address Space, Swapping, Contiguous Memory Allocation, Paging, Segmentation with Paging. Virtual Memory Management in Unix: Demand paging, Process creation, Page Replacement Algorithms, Allocation of Frames, Thrashing. File-System Interface in UNIX: File

concept, Access Methods, Directory structure, Filesystem Mounting, File sharing, Protection and consistency semantics. File-System Implementation: File-System structure. Directory Implementation, Allocation Methods, Free-space Management, Efficiency and Performance, Recovery. Disk Management: Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, Disk Attachment, stable-storage Implementation

UNIT V PROTECTION AND SECURITY

Protection: Goals of Protection, Domain of Protection, Security: Security Problem, User Authentication, One – Time Password, Program Threats, System Threats.

TEXT BOOKS:

1. Operating System Concepts and design by Milan Milonkovic, II Edition, McGraw Hill 1992.
2. Operation System Concepts by Tanenbaum, 2nd Edition, Pearson Education.
3. Operating System by Silberschatz / Galvin / Gagne, 6th Edition,WSE (WILEY Publication)

REFERENCES:

1. Operating System by William Stallings, 4th Edition, Pearson Education.
2. Operating System by H.M.Deitel , 2nd Edition,Pearson Education
3. Operating System by Abraham Silberschatz and peter Baer Galvin, 8th Edition, Pearson Education 1989
4. Operating Systems by Nutt, 3/e Pearson Education 2004

C8 : OOPS With JAVA

OBJECTIVES:
- Object oriented programming is the most proven technique for developing reliable programs.
- It helps in increased productivity, reusability of code, decrease in the development time, and reduces cost of production to an extent. The cost of maintaining such systems have also considerably decreased.
- There are many languages which used the object-oriented concepts and techniques. Some of them are C++, Java, Smalltalk, Objective-C, etc. Java is a purely object-oriented language.
- Systems/applications created using java programming language reduces the need for developing and maintain complex and space consuming applications.

OUTCOMES:
- Students will learn how to write Programs using Java
- Student will learn Java has a lot of advantages of being simple, robust, platform independent, etc. Nowadays java is also found in the mobile phones.
- Student will focuses on the concepts of object oriented programming language and the different constructs for creating applications in java.
- Student will able to learn the object-oriented concepts which helps in the field of programming, management of data, etc.
- Student will get help of Java programming which helps to explore object-oriented nature of the language and the multi-platform versatility offered by it.

UNIT I: Introduction

History, Overview of Java, Object Oriented Programming, A simple Program, two control statements - if statement, for loop, using Blocks of codes, Lexical issues - White space, identifiers, Literals, comments, separators, Java Key words. Data types: Integers, Floating point, characters, Boolean, A closer look at Literals, Variables, Type conversion and casting, Automatic type promotion in Expressions Arrays. Operators: Arithmetic operators, The Bit wise operators, Relational Operators, Boolean Logical operators, Assignment Operator, Operator Precedence. Control Statements: Selection Statements - if, Switch: Iteration Statements - While, Do-while, for Nested loops, Jump statements.

UNIT II: Classes:

Class Fundamentals, Declaring objects, Assigning object reference variables, Methods, constructors, "this" keyword, finalize ( ) method A stack class, Over loading methods, using objects as parameters, Argument passing, Returning objects, Recursion, Access control, Introducing final, understanding static,Introducing Nested and Inner classes, Using command line arguments. Inheritance: Inheritance basics, Using super, method overriding, Dynamic method Dispatch, using abstract classes, using final with Inheritance.

UNIT III: Packages.

Definition, Access protection importing packages, Interfaces: Definition implementing interfaces. Exception Handling: Fundamental, Exception types, Using try and catch, Multiple catch clauses, Nested try Statements, throw, throws, finally, Java's Built - in exception, using Exceptions.

UNIT IV: Multithreaded Programming.

The Java thread model, The main thread, Creating a thread, Creating multiple thread, Creating a thread, Creating multiple threads, Using isalive() and Join(), Thread - Priorities, Synchronization, Inter thread communication, suspending, resuming and stopping threads, using multi-threading. 1/0 basics, Reading control input, writing control output, Reading and Writing files, Applet Fundamentals, the AWT package,AWT Event handling concepts The transient and volatile modifiers, using instance of using assert.

UNIT V: JAVA Database Connectivity (JDBC)

Database connectivity: JDBC architecture, JDBC Drivers, the JDBC API: loading a driver, connecting to a database, Creating and executing JDBC statements, Handling SQL exceptions, Accessing result sets: Types of result sets, Methods of result set interface. An example JDBC application to query a database

TEXT BOOKS:

Java –2: V Edition By Herbert Schildt Pub. TMH.

REFERENCES:

1. SAMS teach yourself Java – 2: 3rd Edition by Rogers Cedenhead and Leura Lemay Pub. Pearson Education.

LIST OF PROGRAMS:
1. Write a program to check whether two strings are equal or not.
2. Write a program to display reverse string.
3. Write a program to find the sum of digits of a given number.
4. Write a program to display a multiplication table.
5. Write a program to display all prime numbers between 1 to 1t000.
6. Write a program to insert element in existing array.
7. Write a program to sort existing array.
8. Write a program to create object for Tree Set and Stack and use all methods.
9. Write a program to check all math class functions.
10. Write a program to execute any Windows 95 application (Like notepad, calculator etc)
11. Write a program to find out total memory, free memory and free memory after executing garbage Collector (gc).
12. Write a program to copy a file to another file using Java to package classes. Get the file names at run time and if the target file is existed then ask confirmation to overwrite and take necessary actions.

13. Write a program to get file name at runtime and display number f lines and words in that file.
14. Write a program to list files in the current working directory depending upon a given pattern.
15. Create a textfileld that allows only numeric value and in specified length.
16. Create a Frame with 2 labels, at runtime display x and y command-ordinate of mouse pointer in the labels.

R PROGRAMMING

OBJECTIVES
- Master the use of the R and RStudio interactive environment.
- Expand r by installing r packages.
- Learn to understand r operator and r functions
- Read structured data into r from various sources.
- Understand the different data types, lists and frames in r.
- Understand the r programming structures.

COURSE OUTCOMES
At the end of the course the students will be able to
1. Learn about R fundamentals
2. Know to implement R operator and R functions
3. Learn to work with Lists and Frames
4. Be able to work with Tables
5. Know about basic Programming Structures in R

UNIT 1 – Introduction.
R – OVERVIEW-Evolution of R -Features of R 2. R – ENVIRONMENT SETUP - Local Environment Setup BASIC SYNTAX -R Command Prompt - Script File - R – DATA TYPES -Vectors -Lists -Matrices -ArraysFactors -Data Frames - R – VARIABLES -Variable Assignment -Data Type of a Variable Finding Variables - Deleting Variables

UNIT 2 – OPERATORS
R – OPERATORS -Types of Operators -Arithmetic Operators-Relational Operators-Logical Operators Assignment Operators-Miscellaneous Operators - R – DECISION MAKING -R - If Statement-R – If...Else Statement -The if...else if...else -Switch Statement - R – LOOP-R - Repeat LoopR - While Loop -R – For Loop -Loop Control Statements-R-Break statement -R – Next Statement . R – FUNCTION -Function Definition -Function Components -Built-in Function –User-defined Function - Calling a Function -Lazy Evaluation of Function –User-defined Function -Calling a Function

UNIT – 3 LISTS AND FRAMES
LISTS- LISTS -Creating a -Naming List Elements - Accessing List Elements - Manipulating List Elements -Merging Lists Converting List to Vector - R – MATRICES -Accessing Elements of a Matrix - Matrix Computation-ARRAYS - Naming Columns and Rows -Accessing Array Elements-Manipulating Array Elements - R – FACTORS -Factors in Data Frame -Changing the Order of Levels -Generating Factor Levels 16. R – DATA FRAMES -Extract Data from Data Frame

UNIT – 4 : FACTORS AND TABLES
Common Functions Used with Factors- The tapply() Function - The split() Function -

The by() Function - Working with Tables- Matrix/Array-Like Operations on Tables- Extended Example: Extracting a Subtable- Extended Example: Finding the Largest Cells in a Table- Table-Related Functions- The aggregate() Function- The cut() Function

UNIT – 5 R PROGRAMMING STRUCTURES

Control Statements- Loops- Looping Over Nonvector Sets - if-else- Arithmetic and Boolean Operators and Values- Default Values for Argument- Return Values- Deciding Whether to Explicitly Call return() - Returning Complex Object- Functions Are Objects..

TEXT BOOKS

1. Matloff, Norman. The art of R programming: A tour of statistical software design. No Starch Press, 2011.

REFERENCE BOOKS

1. Crawley, Michael J. The R book. John Wiley & Sons, 2012.

E BOOKS

1. https://www.cs.upc.edu/

MOOC

1. R Programming Coursera –Johns Hopkins university

BASICS OF R LAB

1) Installing R and R studio
2) Basic operations in r
3) Getting data into R, Basic data manipulation, Loading Data into R
4) Basic plotting
5) Loops and functions
6) Create Vectors, Lists, Arrays, Matrices, Data frames and operations on them.
7) Demonstrate the visualization and graphics using visualization packages.
8) Implement Loop functions with lappy(), sapply(), tapply(), apply(), mapply().
9) Explore data using Single Variables: Unimodal, Bimodal, Histograms, Density Plots, Bar charts
10) Explore data using two Variables: Line plots, Scatter Plots, smoothing cures, Bar charts
11) Explore and implement commands usinfdplyr package
12) Generate random numbers and set seed

SRI VENKATESWARA UNIVERSITY

BCA DEGREE COURSE IN ARTIFICIAL INTELLIGNECE &DATA SCIENCE

SECOND YEAR – THIRD SEMESTER
(Syllabus under CBCS w.e.f. 2021-22)
MODEL QUESTION PAPER (for all papers)

Time: 3 hours   Marks: 75 marks

Note: This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer any five of the following questions in Part A.

Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks

PART – A

Answer any _Five_ of the following question.   **(5X5=25M)**

1	
2	
3	
4	
5	
6	
7	
8	

PART – B
Answer All The Questions. Each question carries 10 marks   **(5X10= 50M)**

9	(A)	
		OR
	(B)	
10	(A)	
		OR
	(B)	
11	(A)	
		OR
	(B)	
12	(A)	
		OR
	(B)	
13	(A)	
		OR
	(B)	