

B.Tech. - Course Structure & Syllabus – RK24**INDUCTION PROGRAMME**

S.No.	Category	Course Name	L-T-P-C
1	MC	Physical Activities -- Sports, Yoga and Meditation, Plantation	0-0-6-0
2	MC	Career Counselling	2-0-2-0
3	MC	Orientation to all branches -- career options, tools, etc.	3-0-0-0
4	EC	Orientation on admitted Branch -- corresponding labs, tools and platforms	2-0-3-0
5	ES	Proficiency Modules & Productivity Tools	2-1-2-0
6	MC	Assessment on basic aptitude and mathematical skills	2-0-3-0
7	MC	Remedial Training in Foundation Courses	2-1-2-0
8	MC	Human Values & Professional Ethics	3-0-0-0
9	BS	Communication Skills -- focus on Listening, Speaking, Reading, Writing skills	2-1-2-0
10	ES	Concepts of Programming	2-0-2-0

Group-A Branches:

Computer Science and Engineering
Computer Science and Engineering (Data Science)

Group-B Branches:


Civil Engineering,
Electrical and Electronics Engineering,
Mechanical Engineering
Electronics and Communication Engineering,
Artificial Intelligence and Machine Learning

Course Code Format:

Regulation	Branch	Year	Semester	Course Serial Number
2 Digits	2 Digits	1 Digit	1 Digit	2 Digits
24	01/02/03/04/05/44/61	1/2/3/4	1/2	01/02/03/04/...../19/20


Branch Code:

Code	Branch
01	Civil Engineering
02	Electrical & Electronics Engineering
03	Mechanical Engineering
04	Electronics and Communication Engineering
05	Computer Science and Engineering
44	Computer Science and Engineering (Data Science)
61	Artificial Intelligence and Machine Learning


Chairman - BoS
Computer Science and Engineering
RKCE (Autonomous)

B. Tech. – I Year I Semester (for Group-A Branches CSE and CSE-(DS))

S.No.	Course Code	Category	Title	L/D	T	P	Credits
1	24051101/ 24441101	BS & H	Engineering Chemistry	3	0	0	3
2	24051102/ 24441102	BS & H	Linear Algebra & Calculus	3	0	0	3
3	24051103/ 24441103	Engineering Science	Basic Civil & Mechanical Engineering	3	0	0	3
4	24051104/ 24441104	BS & H	Communicative English	2	0	0	2
5	24051105/ 24441105	Engineering Science	Introduction to Programming	3	0	0	3
6	24051106/ 24441106	BS & H	Engineering Chemistry Lab	0	0	2	1
7	24051107/ 24441107	Engineering Science	Engineering Workshop	0	0	3	1.5
8	24051108/ 24441108	BS & H	Communicative English Lab	0	0	2	1
9	24051109/ 24441109	Engineering Science	Computer Programming Lab	0	0	3	1.5
10	24051110/ 24441110	BS & H	Health and wellness, Yoga and Sports	-	-	1	0.5
Total				14	0	11	19.5


 Chairman - BOS
 Computer Science and Engineering
 RKCE (Autonomous)


B.Tech. – I Year I Semester (for Group-B Branches CE, EEE, ME, ECE, and AI & ML)

S.No.	Course Code	Category	Title	L/D	T	P	Credits
1	24011111/ 24021111/ 24031111/ 24041111/ 24611111	BS & H	Engineering Physics	3	0	0	3
2	24011112/ 24021112/ 24031112/ 24041112/ 24611112	BS & H	Linear Algebra & Calculus	3	0	0	3
3	24011113/ 24021113/ 24031113/ 24041113/ 24611113	Engineering Science	Basic Electrical & Electronics Engineering	3	0	0	3
4	24011114/ 24021114/ 24031114/ 24041114/ 24611114	Engineering Science	Engineering Drawing	1	0	4	3
5	24011115/ 24021115/ 24031115/ 24041115/ 24611115	Engineering Science	Introduction to Programming	3	0	0	3
6	24011116/ 24021116/ 24031116/ 24041116/ 24611116	BS & H	Engineering Physics Lab	0	0	2	1
7	24011117/ 24021117/ 24031117/ 24041117/ 24611117	Engineering Science	Electrical & Electronics Engineering Workshop	0	0	3	1.5
8	24011118/ 24021118/ 24031118/ 24041118/ 24611118	Engineering Science	IT Workshop	0	0	2	1
9	24011119/ 24021119/ 24031119/ 24041119/ 24611119	Engineering Science	Computer Programming Lab	0	0	3	1.5
10	24011120/ 24021120/ 24031120/ 24041120/ 24611120		NSS / NCC / Scouts & Guides / Community Service	-	-	1	0.5
Total				13	0	15	20.5

ky
Chairman - BOS
Computer Science and Engineering
RKCE (Autonomous)

B.Tech. – I Year II Semester (for Group-A Branches CSE and CSE-(DS))

S.No.	Course Code	Category	Title	L/D	T	P	Credits
1	24051201/ 24441201	BS & H	Engineering Physics	3	0	0	3
2	24051202/ 24441202	BS & H	Differential Equations & Vector Calculus	3	0	0	3
3	24051203/ 24441203	Engineering Science	Basic Electrical & Electronics Engineering	3	0	0	3
4	24051204/ 24441204	Engineering Science	Engineering Drawing	1	0	3	4
5	24051205/ 24441205	Professional Core	Data structures	3	0	0	3
6	24051206/ 24441206	BS & H	Engineering Physics Lab	0	0	2	1
7	24051207/ 24441207	Engineering Science	Electrical & Electronics Engineering Workshop	0	0	3	1.5
8	24051208/ 24441208	Engineering Science	IT Workshop	0	0	2	1
9	24051209/ 24441209	Professional Core	Data structures Lab	0	0	3	1.5
10	24051210/ 24441210		NSS/NCC/Scouts & Guides/Community Service	-	-	1	0.5
Total				13	0	15	20.5


 Chairman - BoS
 Computer Science and Engineering
 RKCE (Autonomous)

Chairman - BoS
 Computer Science and Engineering
 RKCE (Autonomous)

B.Tech. – I Year II Semester (for Group-B Branches CE, EEE, ME, ECE, and AI & ML)


S.No.	Course Code	Category	Title	L/D	T	P	Credits
1	24011211/ 24021211/ 24031211/ 24041211/ 24611211	BS & H	Engineering Chemistry	3	0	0	3
2	24011212/ 24021212/ 24031212/ 24041212/ 24611212	BS & H	Differential Equations & Vector Calculus	3	0	0	3
3	24011213/ 24021213/ 24031213/ 24041213/ 24611213	Engineering Science	Basic Civil & Mechanical Engineering	3	0	0	3
4	24011214/ 24021214/ 24031214/ 24041214/ 24611214	BS & H	Communicative English	2	0	0	2
5	24011215/ 24031215	Professional Core	Engineering Mechanics	3	0	0	3
	24021215		Electrical Circuit Analysis – I				
	24041215		Network Analysis				
	24611215		Data structures				
6	24011216/ 24021216/ 24031216/ 24041216/ 24611216	BS & H	Engineering Chemistry Lab	0	0	2	1
7	24011217/ 24021217/ 24031217/ 24041217/ 24611217	Engineering Science	Engineering Workshop	0	0	3	1.5
8	24011218/ 24021218/ 24031218/ 24041218/ 24611218	BS & H	Communicative English Lab	0	0	2	1
9	24011219/ 24031219	Professional Core	Engineering Mechanics Lab	0	0	3	1.5
	24021219		Electrical Circuits Lab				
	24041219		Network Analysis Lab				
	24611219		Data structures Lab				
10	24011220/ 24021220/ 24031220/ 24041220/ 24611220	BS & H	Health and wellness, Yoga and Sports	-	-	1	0.5
Total				14	0	11	19.5

B. Tech. – II Year I Semester

S.No.	Course Code	Category	Title	L	T	P	Credits
1	24052101	BS&H	Engineering Mathematics (Branch specific)	3	0	0	3
2	24052102	BS&H	Universal Human Values – Understanding Harmony	2	1	0	3
3	24052103	Engineering Science		2	0	0	2
4	24052104	Professional Core		3	0	0	3
5	24052105	Professional Core		3	0	0	3
6	24052106	Engineering Science		0	0	2	1
7	24052107	Professional Core		0	0	3	1.5
8	24052108	Professional Core		0	0	3	1.5
9	24052109	Skill Enhancement course		0	1	2	2
10	24052110	Audit Course	Environmental Science	2	0	0	-
Total				15	2	10	20

B. Tech. – II Year II Semester

S.No.	Course Code	Category	Title	L	T	P	Credits
1	24052201	Management Course - I		2	0	0	2
2	24052202	Engineering Science		3	0	0	3
3	24052203	Professional Core		3	0	0	3
4	24052204	Professional Core		3	0	0	3
5	24052205	Professional Core		3	0	0	3
6	24052206	Professional Core		0	0	2	1
7	24052207	Professional Core		0	0	3	1.5
8	24052208	Professional Core		0	0	3	1.5
9	24052209	Skill Enhancement course		0	1	2	2
10	24052210	BS&H	Design Thinking & Innovation	1	0	2	2
Total				15	1	12	22
Mandatory Community Service Project Internship of 08 weeks duration during summer vacation							



 Chairman - BOS
 Computer Science and Engineering
 RKCE (Autonomous)

B.Tech. – III Year I Semester

S.No.	Course Code	Category	Title	L	T	P	Credits
1	24053101	Professional Core		3	0	0	3
2	24053102	Professional Core		3	0	0	3
3	24053103	Professional Elective - I		2	0	0	2
4	24053104	Open Elective - I		3	0	0	3
5	24053105	Open Elective - II		3	0	0	3
6	24053106	Professional Core		0	0	3	1.5
7	24053107	Professional Core		0	0	3	1.5
8	24053108	Skill Enhancement course		0	1	2	2
9	24053109	BS&H	Tinkering Lab	0	0	2	1
10	24053110	Evaluation of Community Service Internship		-	-	-	2
Total				14	1	10	22

B.Tech. – III Year II Semester

S.No.	Course Code	Category	Title	L	T	P	Credits
1	24053201	Professional Core		3	0	0	3
2	24053202	Professional Core		3	0	0	3
3	24053203	Professional Core		3	0	0	3
4	24053204	Professional Elective - II		3	0	0	3
5	24053205	Professional Elective - III		2	0	0	2
6	24053206	Open Elective - III		3	0	0	3
7	24053207	Professional Core		0	0	2	1
8	24053208	Professional Core		0	0	2	1
9	24053209	Skill Enhancement course		0	1	2	2
10	24053210	Audit Course	Technical Paper Writing & IPR	2	0	0	-
Total				19	1	06	21
Mandatory Industry Internship of 08 weeks duration during summer vacation							

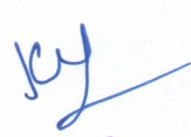

 Chairman - BoS
 Computer Science and Engineering
 RKCE (Autonomous) 7

B.Tech. – IV Year I Semester

S.No.	Course Code	Category	Title	L	T	P	Credits
1	24054101	Professional Core		3	0	0	3
2	24054102	Professional Core		3	0	0	3
3	24054103	Management Course - II		2	0	0	2
4	24054104	Professional Elective - IV		3	0	0	3
5	24054105	Professional Elective - V		3	0	0	3
6	24054106	Open Elective - IV		3	0	0	3
7	24054107	Professional Core		0	0	2	1
8	24054108	Professional Core		0	0	2	1
9	24054109	Skill Enhancement Course		0	1	2	2
10	24054110	Audit Course	Constitution of India	2	0	0	-
11	24054111	Internship	Evaluation of Industry Internship	-	-	-	2
Total				19	1	06	23

B.Tech. – IV Year II Semester

S.No.	Course Code	Category	Title	L	T	P	Credits
1	24054201	Project Work	Full semester Project Work	0	0	12	6
2	24054202	Internship	Full semester Internship	0	0	12	6
Total				0	0	24	12


 Chairman - BOS
 Computer Science and Engineering
 RKCE (Autonomous)

L/D	T	P	C
3	0	0	3

INTRODUCTION TO PROGRAMMING (Common to all Branches of Engineering)

Course Objectives:

- To introduce students to the fundamentals of computer programming.
- To provide hands-on experience with coding and debugging.
- To foster logical thinking and problem-solving skills using programming.
- To familiarize students with programming concepts such as data types, control structures, functions, and arrays.
- To encourage collaborative learning and teamwork in coding projects.

Course Outcomes: A student after completion of the course will be able to

CO1	Understand basics of computers, the concept of algorithm and algorithmic thinking.
CO2	Analyse a problem and develop an algorithm to solve it.
CO3	Implement various algorithms using the C programming language.
CO4	Understand more advanced features of C language. Develop problem-solving skills and the ability to debug and optimize the code.

UNIT I

Introduction to Programming and Problem Solving: History of Computers, Basic organization of a computer: ALU, input-output units, memory, program counter, Introduction to Programming Languages, Basics of a Computer Program Algorithms, flowcharts (Using Dia Tool), pseudo code. Introduction to Compilation and Execution, Primitive Data Types, Variables, and Constants, Basic Input and Output, Operations, Type Conversion, and Casting. **Problem solving techniques:** Algorithmic approach, characteristics of algorithm, **Problem solving strategies:** Top-down approach, Bottom-up approach, Time and space complexities of algorithms.

UNIT II

Control Structures and Arrays: Simple sequential programs Conditional Statements (if, if-else, switch), Loops (for, while, do while) Break and Continue. Arrays indexing, memory model, programs with array of integers, two dimensional arrays.

UNIT III

Strings, Pointers & User Defined Data types: Introduction to Strings, Pointers, dereferencing and address operators, pointer and address arithmetic, array manipulation using pointers, User-defined data types-Structures and Unions.

UNIT IV

Functions & File Handling: Introduction to Functions, Function Declaration and Definition, Function call Return Types and Arguments, modifying parameters inside functions using pointers, arrays as parameters. Scope and Lifetime of Variables, Basics of File Handling

Note: The syllabus is designed with C Language as the fundamental language of implementation.

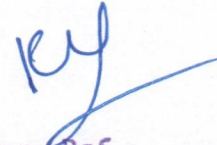
Textbooks:

1. "The C Programming Language", Brian W. Kernighan and Dennis M. Ritchie, PrenticeHall, 1988
2. Schaum's Outline of Programming with C, Byron S Gottfried, McGraw-Hill

Education, 1996

Reference Books:

1. Computing fundamentals and C Programming, Balagurusamy, E., McGraw-Hill Education, 2008.
2. Programming in C, Rema Theraja, Oxford, 2016, 2nd edition
3. C Programming, A Problem Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE, 3rd edition.



**Chairman - BOS
Computer Science and Engineering
RKCE (Autonomous)**

L/D	T	P	C
0	0	3	1.5

COMPUTER PROGRAMMING LAB (Common to all Branches of Engineering)

Course Objectives: The course aims to give students hands – on experience and train them on the concepts of the C- programming language.

Course Outcomes:

CO1	Read, understand, and trace the execution of programs written in C language
CO2	Select the right control structure for solving the problem
CO3	Develop C programs which utilize memory efficiently using programming constructs like pointers
CO4	Develop, Debug and Execute programs to demonstrate the applications of arrays, functions, basic concepts of pointers in C

UNIT 1

WEEK 1

Objective: Getting familiar with the programming environment on the computer and writing the first program.

Suggested Experiments/Activities:

Tutorial 1: Problem-solving using Computers.

Lab1: Familiarization with programming environment

- i) Basic Linux environment and its editors like Vi, Vim & Emacs etc.
- ii) Exposure to Turbo C, gcc
- iii) Writing simple programs using printf(), scanf()

WEEK 2

Objective: Getting familiar with how to formally describe a solution to a problem in a series of finite steps both using textual notation and graphic notation.

Suggested Experiments /Activities:

Tutorial 2: Problem-solving using Algorithms and Flow charts.

Lab 1: Converting algorithms/flow charts into C Source code.

Developing the algorithms/flowcharts for the following sample programs

- i) Sum and average of 3 numbers
- ii) Conversion of Fahrenheit to Celsius and vice versa
- iii) Simple interest calculation

WEEK 3

Objective: Learn how to define variables with the desired data-type, initialize them with appropriate values and how arithmetic operators can be used with variables and constants.

Suggested Experiments/Activities:

Tutorial 3: Variable types and type conversions:

Lab 3: Simple computational problems using arithmetic expressions.

- i) Finding the square root of a given number
- ii) Finding compound interest
- iii) Area of a triangle using heron's formulae
- iv) Distance travelled by an object

UNIT II

WEEK 4

Objective: Explore the full scope of expressions, type-compatibility of variables & constants and operators used in the expression and how operator precedence works.

Suggested Experiments/Activities:

Tutorial4: Operators and the precedence and as associativity:

Lab4: Simple computational problems using the operator' precedence and associativity

- i) Evaluate the following expressions.
 - a. $A+B*C+(D*E) + F*G$
 - b. $A/B*C-B+A*D/3$
 - c. $A+++B---A$
 - d. $J= (i++) + (++i)$
- ii) Find the maximum of three numbers using conditional operator
- iii) Take marks of 5 subjects in integers, and find the total, average in float

WEEK 5

Objective: Explore the full scope of different variants of “if construct” namely if-else, null-else, if-else if*-else, switch and nested-if including in what scenario each one of them can be used and how to use them. Explore all relational and logical operators while writing conditionals for “if construct”.

Suggested Experiments/Activities:

Tutorial 5: Branching and logical expressions:

Lab 5: Problems involving if-then-else structures.

- i) Write a C program to find the max and min of four numbers using if-else.
- ii) Write a C program to generate electricity bill.
- iii) Find the roots of the quadratic equation.
- iv) Write a C program to simulate a calculator using switch case.
- v) Write a C program to find the given year is a leap year or not.

WEEK 6

Objective: Explore the full scope of iterative constructs namely while loop, do-while loop and for loop in addition to structured jump constructs like break and continue including when each of these statements is more appropriate to use.

Suggested Experiments/Activities:

Tutorial 6: Loops, while and for loops

Lab 6: Iterative problems e.g., the sum of series

- i) Find the factorial of given number using any loop.
- ii) Find the given number is a prime or not.
- iii) Compute sine and cos series
- iv) Checking a number palindrome
- v) Construct a pyramid of numbers.

UNIT III

WEEK 7:

Objective: Explore the full scope of Arrays construct namely defining and initializing 1-D and 2-D and more generically n-D arrays and referencing individual array elements from the defined array. Using integer 1-D arrays, explore search solution linear search.

Suggested Experiments/Activities:

Tutorial 7: 1 D Arrays: searching.

Lab 7: 1D Array manipulation, linear search

- i) Find the min and max of a 1-D integer array.
- ii) Perform linear search on 1D array.
- iii) The reverse of a 1D integer array
- iv) Find 2's complement of the given binary number.
- v) Eliminate duplicate elements in an array.

WEEK 8:

Objective: Explore the difference between other arrays and character arrays that can be used as Strings by using null character and get comfortable with string by doing experiments that will reverse a string and concatenate two strings. Explore sorting solution bubble sort using integer arrays.



Suggested Experiments/Activities:**Tutorial 8:** 2 D arrays, sorting and Strings.**Lab 8:** Matrix problems, String operations, Bubble sort

- i) Addition of two matrices
- ii) Multiplication two matrices
- iii) Sort array elements using bubble sort
- iv) Concatenate two strings without built-in functions
- v) Reverse a string using built-in and without built-in string functions

UNIT IV**WEEK 9:****Objective:** Explore pointers to manage a dynamic array of integers, including memory allocation & value initialization, resizing changing and reordering the contents of an array

and memory de-allocation using malloc (), calloc (), realloc () and free () functions. Gain experience processing command-line arguments received by C

Suggested Experiments/Activities:**Tutorial 9:** Pointers, structures and dynamic memory allocation**Lab 9:** Pointers and structures, memory dereference.

- i) Write a C program to find the sum of a 1D array using malloc()
- ii) Write a C program to find the total, average of n students using structures
- iii) Enter n students data using calloc() and display failed students list
- iv) Read student name and marks from the command line and display the student details along with the total.
- v) Write a C program to implement realloc()

WEEK 10:**Objective:** Experiment with C Structures, Unions, bit fields and self-referential structures (Singly linked lists) and nested structures**Suggested Experiments/Activities:****Tutorial 10:** Bitfields, Self-Referential Structures, Linked lists**Lab10 :** Bitfields, linked lists

Read and print a date using dd/mm/yyyy format using bit-fields and differentiate the same without using bit- fields

- i) Create and display a singly linked list using self-referential structure.
- ii) Demonstrate the differences between structures and unions using a C program.
- iii) Write a C program to shift/rotate using bitfields.
- iv) Write a C program to copy one structure variable to another structure of the same type.

UNIT V**WEEK 11:****Objective:** Explore the Functions, sub-routines, scope and extent of variables, doing some experiments by parameter passing using call by value. Basic methods of numerical integration**Suggested Experiments/Activities:****Tutorial 11:** Functions, call by value, scope and extent,**Lab 11:** Simple functions using call by value, solving differential equations using Eulers theorem.

- i) Write a C function to calculate NCR value.
- ii) Write a C function to find the length of a string.
- iii) Write a C function to transpose of a matrix.
- iv) Write a C function to demonstrate numerical integration of differential equations

using Euler's method

WEEK 12:

Objective: Explore how recursive solutions can be programmed by writing recursive functions that can be invoked from the main by programming at-least five distinct problems that have naturally recursive solutions.

Suggested Experiments/Activities:

Tutorial 12: Recursion, the structure of recursive calls

Lab 12: Recursive functions

- i) Write a recursive function to generate Fibonacci series.
- ii) Write a recursive function to find the lcm of two numbers.
- iii) Write a recursive function to find the factorial of a number.
- iv) Write a C Program to implement Ackermann function using recursion.
- v) Write a recursive function to find the sum of series.

WEEK 13:

Objective: Explore the basic difference between normal and pointer variables, Arithmetic operations using pointers and passing variables to functions using pointers

Suggested Experiments/Activities:

Tutorial 13: Call by reference, dangling pointers

Lab 13: Simple functions using Call by reference, Dangling pointers.

- i) Write a C program to swap two numbers using call by reference.
- ii) Demonstrate Dangling pointer problem using a C program.
- iii) Write a C program to copy one string into another using pointer.
- iv) Write a C program to find no of lowercase, uppercase, digits and other characters using pointers.

WEEK14:

Objective: To understand data files and file handling with various file I/O functions. Explore the differences between text and binary files.

Suggested Experiments/Activities:

Tutorial 14: File handling

Lab 14: File operations

- i) Write a C program to write and read text into a file.
- ii) Write a C program to write and read text into a binary file using fread() and fwrite()
- iii) Copy the contents of one file to another file.
- iv) Write a C program to merge two files into the third file using command-line arguments.
- v) Find no. of lines, words and characters in a file
- vi) Write a C program to print last n characters of a given file.

Textbooks:

1. Ajay Mittal, Programming in C: A practical approach, Pearson.
2. Byron Gottfried, Schaum's Outline of Programming with C, McGraw Hill

Reference Books:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice- Hall of India
2. C Programming, A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE.

L/D	T	P	C
0	0	2	1

IT WORKSHOP (Common to all Branches of Engineering)

Course Objectives:

- To introduce the internal parts of a computer, peripherals, I/O ports, connecting cables
- To demonstrate configuring the system as Dual boot both Windows and other Operating Systems Viz. Linux, BOSS
- To teach basic command line interface commands on Linux.
- To teach the usage of Internet for productivity and self-paced life-long learning
- To introduce Compression, Multimedia and Antivirus tools and Office Tools such as Word processors, Spread sheets and Presentation tools.

Course Outcomes: After completion of this course, the student will be able to

CO1	Perform Hardware troubleshooting
CO2	Understand Hardware components and inter dependencies
CO3	Safeguard computer systems from viruses/worms
CO4	Document/ Presentation preparation
CO5	Perform calculations using spreadsheets

PC Hardware & Software Installation

Task 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Task 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Task 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Task 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot (VMWare) with both Windows and Linux. Lab instructors should verify the installation and follow it up with a Viva

Task 5: Every student should install BOSS on the computer. The system should be configured as dual boot (VMWare) with both Windows and BOSS. Lab instructors should verify the installation and follow it up with a Viva

Internet & World Wide Web

Task1: Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

Task 2: Web Browsers, Surfing the Web: Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

Task 3: Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

Task 4: Cyber Hygiene: Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

LaTeX and WORD

Task 1 – Word Orientation: The mentor needs to give an overview of La TeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of La TeX and MS office or equivalent (FOSS) tool Word as word Processors, Details of the four tasks and features that would be covered in each, Using La TeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

Task 2: Using La TeX and Word to create a project certificate. Features to be covered:- Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in both La TeX and Word.

Task 3: Creating project abstract Features to be covered:-Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

Task 4: Creating a Newsletter: Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

EXCEL

Excel Orientation: The mentor needs to tell the importance of MS office or equivalent (FOSS) tool Excel as a Spreadsheet tool, give the details of the four tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

Task 1: Creating a Scheduler - Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text

Task 2: Calculating GPA -. Features to be covered:- Cell Referencing, Formulae in excel – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function,

LOOKUP/VLOOKUP

Task 3: Split cells, freeze panes, group and outline, Sorting, Boolean and logical operators, Conditional formatting

POWER POINT

Task 1: Students will be working on basic power point utilities and tools which help them create basic power point presentations. PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in PowerPoint.

Task 2: Interactive presentations - Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts.

Task 3: Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides.

AI TOOLS – ChatGPT

Task 1: Prompt Engineering: Experiment with different types of prompts to see how the model responds. Try asking questions, starting conversations; or even providing incomplete sentences to see how the model completes them.

• Ex: Prompt: "You are a knowledgeable AI. Please answer the following question: What is the capital of France?"

Task 2: Creative Writing: Use the model as a writing assistant. Provide the beginning of a

story or a description of a scene, and let the model generate the rest of the content. This can be a fun way to brainstorm creative ideas

- Ex: Prompt: "In a world where gravity suddenly stopped working, people started floating upwards. Write a story about how society adapted to this new reality."

Task 3: Language Translation: Experiment with translation tasks by providing a sentence in one language and asking the model to translate it into another language. Compare the output to see how accurate and fluent the translations are.

- Ex: Prompt: "Translate the following English sentence to French: 'Hello, how are you doing today?'"

Reference Books:

1. Comdex Information Technology course tool kit, Vikas Gupta, WILEY Dream tech, 2003
2. The Complete Computer upgrade and repair book, Cheryl A. Schmidt, WILEY Dream tech, 2013, 3rd edition
3. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education, 2012, 2nd edition
4. PC Hardware - A Handbook, Kate J. Chase, PHI (Microsoft)
5. LaTeX Companion, Leslie Lamport, PHI/Pearson.
6. IT Essentials PC Hardware and Software Companion Guide, David Anfinson and Ken Quamme. – CISCO Press, Pearson Education, 3rd edition
7. IT Essentials PC Hardware and Software Labs and Study Guide, Patrick Regan– CISCO Press, Pearson Education, 3rd edition



Chairman - BoS
Computer Science and Engineering
RKCE (Autonomous)

L/D	T	P	C
3	0	0	3

DATA STRUCTURES

(Common to Computer Science Engineering and Allied Branches)

Course Objectives:

- To provide the knowledge of basic data structures and their implementations.
- To understand importance of data structures in context of writing efficient programs.
- To develop skills to apply appropriate data structures in problem solving.

Course Outcomes: At the end of the course, Student will be able to

CO1	Explain the role of linear data structures in organizing and accessing data efficiently in algorithms
CO2	Develop programs using stacks to handle recursive algorithms, manage program states, and solve related problems. Apply queue-based algorithms for efficient task scheduling and breadth-first traversal in graphs and distinguish between dequeues and priority queues, and apply them appropriately to solve data management challenges
CO3	Design, implement, and apply linked lists for dynamic data storage, demonstrating understanding of memory allocation
CO4	Devise novel solutions to small scale programming challenges involving data structures such as stacks, queues, Trees. Recognize scenarios where hashing is advantageous, and design hash-based solutions for specific problems

UNIT I

Introduction to Linear Data Structures: Definition and importance of linear data structures, Abstract data types (ADTs) and their implementation, Overview of time and space complexity analysis for linear data structures. Searching Techniques: Linear & Binary Search, Sorting Techniques: Bubble sort, Selection sort, Insertion Sort.

UNIT II

Stacks: Introduction to stacks: properties and operations, implementing stacks using arrays and linked lists, Applications of stacks in expression evaluation, backtracking, reversing list etc.

Queues: Introduction to queues: properties and operations, implementing queues using arrays and linked lists, Applications of queues in breadth-first search, scheduling, etc.

Dequeues: Introduction to dequeues (double-ended queues), Operations on dequeues and their applications.

UNIT III

Linked Lists: Singly linked lists: representation and operations, doubly linked lists and circular linked lists, Comparing arrays and linked lists, Applications of linked lists.

UNIT IV

Trees: Introduction to Trees, Binary Search Tree – Insertion, Deletion & Traversal Hashing: Brief introduction to hashing and hash functions, Collision resolution techniques: chaining and open addressing, Hash tables: basic implementation and operations, Applications of hashing in unique identifier generation, caching, etc.

Textbooks:

1. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2nd Edition.
2. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan AndersonFreed, Silicon Press, 2008

Reference Books:

1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders
2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft
3. Problem Solving with Algorithms and Data Structures" by Brad Miller and David Ranum
4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms" by Robert Sedgewick



**Chairman - BOS
Computer Science and Engineering
RKCE (Autonomous)**

L/D	T	P	C
0	0	3	1.5

DATA STRUCTURES LAB
(Common to Computer Science Engineering and Allied Branches)

Course Objectives:

The course aims to strengthen the ability of the students to identify and apply the suitable data structure for the given real-world problem. It enables them to gain knowledge in practical applications of data structures.

Course Outcomes: At the end of the course, Student will be able to

CO1	Explain the role of linear data structures in organizing and accessing data efficiently in algorithms
CO2	Design, implement, and apply linked lists for dynamic data storage, demonstrating understanding of memory allocation
CO3	Develop programs using stacks to handle recursive algorithms, manage program states, and solve related problems
CO4	Apply queue-based algorithms for efficient task scheduling and breadth-first traversal in graphs and distinguish between dequeues and priority queues and apply them appropriately to solve data management challenges
CO5	Recognize scenarios where hashing is advantageous, and design hash-based solutions for specific problems

List of Experiments:**Exercise 1: Array Manipulation**

- i. Write a program to reverse an array.
- ii. C Programs to implement the Searching Techniques – Linear & Binary Search
- iii. C Programs to implement Sorting Techniques – Bubble, Selection and Insertion Sort

Exercise 2: Linked List Implementation

- i. Implement a singly linked list and perform insertion and deletion operations.
- ii. Develop a program to reverse a linked list iteratively and recursively.
- iii. Solve problems involving linked list traversal and manipulation.

Exercise 3: Linked List Applications

- i. Create a program to detect and remove duplicates from a linked list.
- ii. Implement a linked list to represent polynomials and perform addition.
- iii. Implement a double-ended queue (deque) with essential operations.

Exercise 4: Double Linked List Implementation

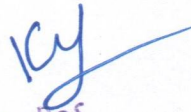
- i. Implement a doubly linked list and perform various operations to understand its properties and applications.
- ii. Implement a circular linked list and perform insertion, deletion, and traversal.

Exercise 5: Stack Operations

- i. Implement a stack using arrays and linked lists.
- ii. Write a program to evaluate a postfix expression using a stack.
- iii. Implement a program to check for balanced parentheses using a stack.

Exercise 6: Queue Operations

- i. Implement a queue using arrays and linked lists.
- ii. Develop a program to simulate a simple printer queue system.


 Chairman - BOS
 Computer Science and Engineering
 RKCE (Autonomous)

- iii. Solve problems involving circular queues.

Exercise 7: Stack and Queue Applications

- i. Use a stack to evaluate an infix expression and convert it to postfix.
- ii. Create a program to determine whether a given string is a palindrome or not.
- iii. Implement a stack or queue to perform comparison and check for symmetry.

Exercise 8: Binary Search Tree

- i. Implementing a BST using Linked List.
- ii. Traversing of BST.

Exercise 9: Hashing

- i. Implement a hash table with collision resolution techniques.
- ii. Write a program to implement a simple cache using hashing.

Textbooks:

1. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2nd Edition.
2. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Press, 2008

Reference Books:

1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders
2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft
3. Problem Solving with Algorithms and Data Structures" by Brad Miller and David Ranum
4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms by Robert Sedgewick.



**Chairman - BoS
Computer Science and Engineering
RKCE (Autonomous)**