



ELLENKI COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous Institution - UGC, Govt. of India)

(Sponsored by Ellenki Educational Society)

(Approved by AICTE, New Delhi, Affiliated to JNTUH Hyderabad, MSME - HI Govt. of India,
Accredited by NAAC, Recognition of 2(f) by UGC, ISO 9001:2015 Certified)




Date: 01-11-2023

Board of Studies of Computer Science & Engineering Dept.

On behalf of ELLENKI COLLEGE OF ENGINEERING AND TECHNOLOGY (Autonomous), Patelguda, Sangareddy-502319. I am pleased to constitute the Board of Studies in the Department of Computer Science and Engineering for B.Tech and M. Tech Courses as per details given below:

S.NO	NAME	DESIGNATION	DESIGNATION IN BOS
1	Dr. Shaik Mahaboob Sharief	Professor & Head of Dept. of CSE, ECET	Chairman
2	Dr.B.V Ram Naresh Yadav	Professor of CSE Department, JNTUH UCES	Nominee-JNTUH
3	Dr. M.Narayanan	Professor of CSE Department, School of Engineering, Malla Reddy University, Hyderabad	Member Other College
4	Dr. G.Shyama Chandra Prasad	Professor, Matrusri Engineering College, Hyderabad	Member Other College
5	Mr.Rajashekar Gudi	Senior Associate, Data Engineering State street Corporation	Member Industry
6	Dr.S.Swathi Rao	Professor & Head of Dept. of CSM, ECET	Member College
7	Dr.P. Nageswara Rao	Professor & Head of Dept. of CSC, ECET	Member College
8	Dr. E. Srinivas Raju	Professor & Head of Dept. of CSD, ECET.	Member College
9	G.Shailaja	M.Tech, , Alumni	Member Alumni
10	Dr.P.John Paul	Principal, ECET	Special Invitee

- The above staff members of the Board of Studies in Computer Science and Engineering shall hold the office for a period of Three years with effect from the date of issue of this order.
- The members attending the meeting of the Board of Studies are eligible for T.A. and D.A as per rules of the Institution in force.
- The members are also requested to intimate this office in case of any changes in their address and designations.
- We request you to kindly consent your willingness to the member of this BOS.


Principal
Prof. P. John Paul

PRINCIPAL



Department of Computer Science and Engineering

Minutes of Board of Studies Meeting

Date: 16/11/2023

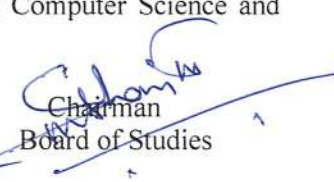
Ellenki College of Engineering & Technology was founded in the year 1999 with a vision to achieve excellence in providing all round education. Established for over two decades, ELLENKI College of Engineering & Technology is one of the premier private engineering colleges in Hyderabad. The College has got Autonomous Status from the A.Y. 2023-24 for 5 years.

The first BOS meeting of the Computer Science and Engineering Department was held on 16th November, 2023 in dual mode. The minutes of meeting are as follows.


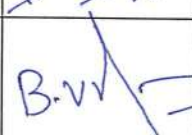



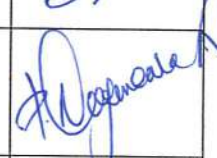


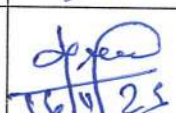
The Chairman welcomed all the members to the 1st Board of Studies meeting Computer Science and Engineering Department.

1. The Academic course structure for B.Tech-CSE (I, II, III & IV year) has been discussed and drafted for ER23 Regulations.
2. Detailed syllabi for the B.Tech-CSE (I Year) program have been discussed at length. The proposed syllabus has been agreed and no changes were suggested by the BOS members.
3. The proposed syllabus for Programming for problem solving, Elements of computer science and engineering offering to CSE, CSE (AI&ML), CSE (DS), and CSE (CS) in I B. Tech I Semester has been approved.
4. The proposed syllabus for Programming for problem solving Laboratory for CSE, CSE(AI&ML), CSE(DS), CSE(CS) in I B. Tech I Semester has been approved.
5. The proposed syllabus for C Programming and Data Structures Laboratory for EEE, Mechanical Engineering and Civil Engineering, C Programming for Engineers Laboratory for ECE in I B. Tech I Semester respectively has been approved.
6. The proposed syllabus for Python Programming Laboratory, IT Workshop for CSE, CSE(AI&ML), CSE(DS), CSE(CS) in I B. Tech II Semester has been approved. The proposed syllabus for Python Programming Laboratory for Civil Engineering, Mechanical Engineering, Applied Python Programming Laboratory for EEE and ECE departments in I B. Tech II Semester respectively has been approved.
7. Academic course structure for M.Tech (Computer Science and Engineering), I & II years have been discussed and drafted for ER23 Regulations.
8. Detailed syllabi for M. Tech (Computer Science and Engineering), I Year program have been discussed at length. The proposed syllabus has been agreed upon by the BOS Members.

Finally, the Chairman thanked all the members for their presence and also for their valuable suggestions towards the importance of the Curriculum and Syllabus of Computer Science and Engineering.


Chairman
Board of Studies

**Board of Studies of Computer Science & Engineering Dept.
Attendance Sheet**

S.NO	NAME	DESIGNATION	DESIGNATION IN BOS	Signature
1	Dr. Shaik Mahaboob Sharief	Professor & Head of Dept. of CSE, ECET	Chairman	
2	Dr.B.V Ram Naresh Yadav	Professor of CSE Department, JNTUH UCES	JNTUH Nominee	B.V.  16/11/23
3	Dr. M.Narayanan	Professor of CSE Department, School of Engineering, Malla Reddy University, Hyderabad	Member-Other College	
4	Dr. G.Shyama Chandra Prasad	Professor, Matrusri Engineering College, Hyderabad	Member-Other College	
5	Mr. Rajashekar Gudi	Senior Associate, Data Engineering State street Corporation	Member-Industry	
6	Dr.S.Swathi Rao	Professor & Head of Dept. of CSM, ECET	Member-College	
7	Dr.P. Nageswara Rao	Professor & Head of Dept. of CSC, ECET	Member-College	
8	Dr. E. Srinivas Raju	Professor & Head of Dept. of CSD, ECET.	Member-College	
9	G.Shailaja	M.Tech, , Alumni	Member-Alumni	
10	Dr.P.John Paul	Principal, ECET	Special Invitee	 16/11/23

ELLENKI COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS), HYDERABAD
PROPOSED SYLLABUS

B.Tech. in COMPUTER SCIENCE AND ENGINEERING
COURSE STRUCTURE, I & II YEAR SYLLABUS (ER23 Regulations)
Applicable from AY 2023-24 Batch

I Year I Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA101BS	Matrices and Calculus	3	1	0	4
2.	CH102BS	Engineering Chemistry	3	1	0	4
3.	CS103ES	Programming for Problem Solving	3	0	0	3
4.	EE104ES	Basic Electrical Engineering	2	0	0	2
5.	ME105ES	Computer Aided Engineering Graphics	1	0	4	3
6.	CS106ES	Elements of Computer Science & Engineering	0	0	2	1
7.	CH107BS	Engineering Chemistry Laboratory	0	0	2	1
8.	CS108ES	Programming for Problem Solving Laboratory	0	0	2	1
9.	EE109ES	Basic Electrical Engineering Laboratory	0	0	2	1
		Induction Program				
Total			12	2	12	20

I Year II Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA201BS	Ordinary Differential Equations and Vector Calculus	3	1	0	4
2.	PH202BS	Applied Physics	3	1	0	4
3.	ME203ES	Engineering Workshop	0	1	3	2.5
4.	EN204HS	English for Skill Enhancement	2	0	0	2
5.	EC205ES	Electronic Devices and Circuits	2	0	0	2
6.	CS206ES	Python Programming Laboratory	0	1	2	2
7.	PH207BS	Applied Physics Laboratory	0	0	3	1.5
8.	EN208HS	English Language and Communication Skills Laboratory	0	0	2	1
9.	CS209ES	IT Workshop	0	0	2	1
10.	*MC210	Environmental Science	3	0	0	0
Total			13	4	12	20

II YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Digital Electronics	3	0	0	3
2		Data Structures	3	0	0	3
3		Computer Oriented Statistical Methods	3	1	0	4
4		Computer Organization and Architecture	3	0	0	3
5		Object Oriented Programming through Java	3	0	0	3
6		Data Structures Lab	0	0	3	1.5
7		Object Oriented Programming through Java Lab	0	0	3	1.5
8		Data visualization- R Programming/ Power BI	0	0	2	1
9	*MC	Gender Sensitization Lab	0	0	2	0
Total			15	1	10	20

S. Shankar
Co-Ordinator

B. J.

C. Rajashekhara

S. Raju

II YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Discrete Mathematics	3	0	0	3
2		Business Economics & Financial Analysis	3	0	0	3
3		Operating Systems	3	0	0	3
4		Database Management Systems	3	0	0	3
5		Software Engineering	3	0	0	3
6		Operating Systems Lab	0	0	2	1
7		Database Management Systems Lab	0	0	2	1
8		Real-time Research Project/ Societal Related Project	0	0	4	2
9		Node JS/ React JS/ Django	0	0	2	1
10	*MC	Constitution of India	3	0	0	0
Total			18	0	10	20

III YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Design and Analysis of Algorithms	3	1	0	4
2		Computer Networks	3	0	0	3
3		DevOps	3	0	0	3
4		Professional Elective-I	3	0	0	3
5		Professional Elective -II	3	0	0	3
6		Computer Networks Lab	0	0	2	1
7		DevOps Lab	0	0	2	1
8		Advanced English Communication Skills Lab	0	0	2	1
9		UI design- Flutter	0	0	2	1
10	*MC	Intellectual Property Rights	3	0	0	0
Total			18	1	8	20

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Machine Learning	3	0	0	3
2		Formal Languages and Automata Theory	3	0	0	3
3		Artificial Intelligence	3	0	0	3
4		Professional Elective - III	3	0	0	3
5		Open Elective-I	3	0	0	3
6		Machine Learning Lab	0	0	2	1
7		Artificial Intelligence Laboratory	0	0	2	1
8		Professional Elective-III Lab	0	0	2	1
9		Industrial Oriented Mini Project/ Internship/ Skill Development Course (Big data-Spark)	0	0	4	2
10	*MC	Environmental Science	3	0	0	0
Total			18	0	8	20

S. Sharma
2023

B. L.

G. Rajesh Kumar

[Signature]

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

IV YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Cryptography and Network Security	3	0	0	3
2		Compiler Design	3	0	0	3
3		Professional Elective -IV	3	0	0	3
4		Professional Elective -V	3	0	0	3
5		Open Elective - II	3	0	0	3
6		Cryptography and Network Security Lab	0	0	2	1
7		Compiler Design Lab	0	0	2	1
8		Project Stage - I	0	0	6	3
		Total Credits	15	0	10	20

IV YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Organizational Behavior	3	0	0	3
2		Professional Elective – VI	3	0	0	3
3		Open Elective – III	3	0	0	3
4		Project Stage – II including Seminar	0	0	22	11
		Total Credits	9	0	22	20

*MC – Satisfactory/Unsatisfactory

#Skill Course - 1 credit with 2 Practical Hours

Professional Elective - I

	Quantum Computing
	Advanced Computer Architecture
	Data Analytics
	Image Processing
	Principles of Programming Languages

Professional Elective - II

	Computer Graphics
	Embedded Systems
	Information Retrieval Systems
	Distributed Databases
	Natural Language Processing

Professional Elective - III

	Full Stack Development
	Internet of Things
	Scripting Languages
	Mobile Application Development
	Software Testing Methodologies

Courses in PE - III and PE - III Lab must be in 1-1 correspondence.

S. S. S.

B. S.

C. Rajashekar

A. Jayaram Reddy

Professional Elective -IV

	Graph Theory
	Advanced Operating Systems
	Soft Computing
	Cloud Computing
	Ad hoc & Sensor Networks

Professional Elective -V

	Advanced Algorithms
	Agile Methodology
	Robotic Process Automation
	Blockchain Technology
	Software Process & Project Management

Professional Elective - VI

	Computational Complexity
	Distributed Systems
	Deep Learning
	Human Computer Interaction
	Cyber Forensics

Open Elective -1:

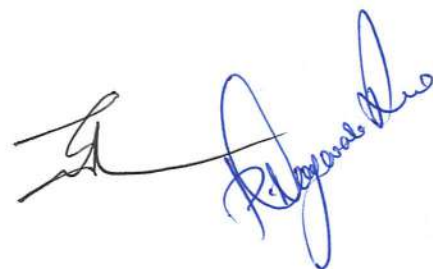
1. Data Structures
2. Database Management Systems

Open Elective -2:

1. Operating Systems
2. Software Engineering

Open Elective -3:

1. Algorithms Design and Analysis
2. Introduction to Computer Networks



PROGRAMMING FOR PROBLEM SOLVING

B.Tech. I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:

- To learn the fundamentals of computers.
- To understand the various steps in program development.
- To learn the syntax and semantics of the C programming language.
- To learn the usage of structured programming approaches in solving problems.

Course Outcomes: The student will learn

- To write algorithms and to draw flowcharts for solving problems.
- To convert the algorithms/flowcharts to C programs.
- To code and test a given logic in the C programming language.
- To decompose a problem into functions and to develop modular reusable code.
- To use arrays, pointers, strings and structures to write C programs.
- Searching and sorting problems.

UNIT - I: Introduction to Programming

Compilers, compiling and executing a program.

Representation of Algorithm - Algorithms for finding roots of a quadratic equations, finding minimum and maximum numbers of a given set, finding if a number is prime number Flowchart/Pseudocode with examples, Program design and structured programming

Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code, Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators

Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto, Iteration with for, while, do- while loops

I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments

UNIT - II: Arrays, Strings, Structures and Pointers:

Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays

Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings

Structures: Defining structures, initializing structures, unions, Array of structures

Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of Pointers in self-referential structures, usage of self referential structures in linked list (no implementation) Enumeration data type

UNIT - III: Preprocessor and File handling in C:

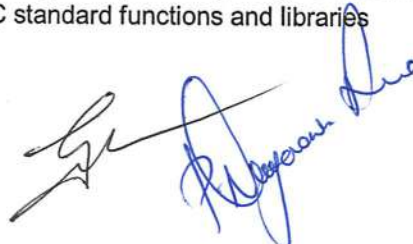
Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef

Files: Text and Binary files, Creating and Reading and writing text and binary files, Appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

UNIT - IV: Function and Dynamic Memory Allocation:

Functions: Designing structured programs, Declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries



Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions
Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

UNIT - V: Searching and Sorting:

Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms), Basic concept of order of complexity through the example programs

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill






C. Rajashekar



ELEMENTS OF COMPUTER SCIENCE AND ENGINEERING**B.Tech. I Year I Sem.**

L	T	P	C
0	0	2	1

Course Objective: To provide an overview of the subjects of computer science and engineering.

Course Outcomes:

1. Know the working principles of functional units of a basic Computer
2. Understand program development, the use of data structures and algorithms in problem solving.
3. Know the need and types of operating system, database systems.
4. Understand the significance of networks, internet, WWW and cyber security.
5. Understand Autonomous systems, the application of artificial intelligence.

UNIT – I

Basics of a Computer – Hardware, Software, Generations of computers. Hardware - functional units, Components of CPU, Memory – hierarchy, types of memory, Input and output devices. Software – systems software, application software, packages, frameworks, IDEs.

UNIT – II

Software development – waterfall model, Agile, Types of computer languages – Programming, markup, scripting Program Development – steps in program development, flowcharts, algorithms, data structures – definition, types of data structures

UNIT – III

Operating systems: Functions of operating systems, types of operating systems, Device & Resource management

Database Management Systems: Data models, RDBMS, SQL, Database Transactions, data centers, cloud services

UNIT – IV

Computer Networks: Advantages of computer networks, LAN, WAN, MAN, internet, WiFi, sensor networks, vehicular networks, 5G communication.

World Wide Web – Basics, role of HTML, CSS, XML, Tools for web designing, Social media, Online social networks.

Security – information security, cyber security, cyber laws

UNIT – V

Autonomous Systems: IoT, Robotics, Drones, Artificial Intelligence – Learning, Game Development, natural language processing, image and video processing.

Cloud Basics

TEXT BOOK:

1. Invitation to Computer Science, G. Michael Schneider, Macalester College, Judith L. Gersting University of Hawaii, Hilo, Contributing author: Keith Miller University of Illinois, Springfield.

REFERENCE BOOKS:

1. Fundamentals of Computers, Reema Thareja, Oxford Higher Education, Oxford University Press.
2. Introduction to computers, Peter Norton, 8th Edition, Tata McGraw Hill.
3. Computer Fundamentals, Anita Goel, Pearson Education India, 2010.
4. Elements of computer science, Cengage.

Handwritten signatures in blue ink.

Handwritten signature in green ink.

Handwritten signature in blue ink.

Handwritten signature in blue ink.

PROGRAMMING FOR PROBLEM SOLVING LABORATORY

B.Tech. I Year I Sem.

L T P C
0 0 2 1

[Note: The programs may be executed using any available Open Source/ Freely available IDE

Some of the Tools available are:

CodeLite: <https://codelite.org/>

Code:Blocks: <http://www.codeblocks.org/>

DevC++ : <http://www.bloodshed.net/devcpp.html>

Eclipse: <http://www.eclipse.org>

This list is not exhaustive and is NOT in any order of preference]

Course Objectives: The students will learn the following:

- To work with an IDE to create, edit, compile, run and debug programs
- To analyze the various steps in program development.
- To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
- To develop modular, reusable and readable C Programs using the concepts like functions, arrays etc.
- To Write programs using the Dynamic Memory Allocation concept.
- To create, read from and write to text and binary files

Course Outcomes: The candidate is expected to be able to:

- formulate the algorithms for simple problems
- translate given algorithms to a working and correct program
- correct syntax errors as reported by the compilers
- identify and correct logical errors encountered during execution
- represent and manipulate data with arrays, strings and structures
- use pointers of different types
- create, read and write to and from simple text and binary files
- modularize the code with functions so that they can be reused

Practice sessions:

- a. Write a simple program that prints the results of all the operators available in C (including pre/post increment, bitwise and/or/not, etc.). Read required operand values from standard input.
- b. Write a simple program that converts one given data type to another using auto conversion and casting. Take the values from standard input.

Simple numeric problems:

- a. Write a program for finding the max and min from the three numbers.
- b. Write the program for the simple, compound interest.
- c. Write a program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input.
- d. Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:
- e. $5 \times 1 = 5$
- f. $5 \times 2 = 10$
- g. $5 \times 3 = 15$
- h. Write a program that shows the binary equivalent of a given positive number between 0 to 255.

S. S. S.

S. S. S.

B. V. S.

A. Rajesh Kumar

A.

A. Rajesh Kumar

Expression Evaluation:

- A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula $s = ut + \frac{1}{2}at^2$ where u and a are the initial velocity in m/sec ($= 0$) and acceleration in m/sec^2 ($= 9.8 \text{ m/s}^2$)).
- Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators $+, -, *, /, \%$ and use Switch Statement)
- Write a program that finds if a given number is a prime number
- Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
- A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
- Write a C program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
- Write a C program to find the roots of a Quadratic equation.
- Write a C program to calculate the following, where x is a fractional value.
 - $1 - x/2 + x^2/4 - x^3/6$
- Write a C program to read in two numbers, x and n , and then compute the sum of this geometric progression: $1 + x + x^2 + x^3 + \dots + x^n$. For example: if n is 3 and x is 5, then the program computes $1 + 5 + 25 + 125$.

Arrays, Pointers and Functions:

- Write a C program to find the minimum, maximum and average in an array of integers.
- Write a function to compute mean, variance, Standard Deviation, sorting of n elements in a single dimension array.
- Write a C program that uses functions to perform the following:
 - Addition of Two Matrices
 - Multiplication of Two Matrices
 - Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be the same.
- Write C programs that use both recursive and non-recursive functions
- To find the factorial of a given integer.
- To find the GCD (greatest common divisor) of two given integers.
- To find x^n
- Write a program for reading elements using a pointer into an array and display the values using the array.
- Write a program for display values reverse order from an array using a pointer.
- Write a program through a pointer variable to sum of n elements from an array.

Files:

- Write a C program to display the contents of a file to standard output device.
- Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
- Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.
- Write a C program that does the following:
It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function)
Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function)
The program should then read all 10 values and print them back.

- e. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

Strings:

- Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
- Write a C program that converts a number ranging from 1 to 50 to Roman equivalent
- Write a C program that uses functions to perform the following operations:
- To insert a sub-string into a given main string from a given position.
- To delete n Characters from a given position in a given string.
- Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)
- Write a C program that displays the position of a character ch in the string S or - 1 if S doesn't contain ch.
- Write a C program to count the lines, words and characters in a given text.

Miscellaneous:

- Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

- Write a C program to construct a pyramid of numbers as follows:

```

1           *           1           1           *
1 2        **          2 3         2 2         **
1 2 3      ***          4 5 6       3 3 3       ***
                                     4 4 4 4      **
                                                         *
```

Sorting and Searching:

- Write a C program that uses non recursive function to search for a Key value in a given
- list of integers using linear search method.
- Write a C program that uses non recursive function to search for a Key value in a given
- sorted list of integers using binary search method.
- Write a C program that implements the Bubble sort method to sort a given list of
- integers in ascending order.
- Write a C program that sorts the given array of integers using selection sort in descending order
- Write a C program that sorts the given array of integers using insertion sort in ascending order
- Write a C program that sorts a given array of names

TEXT BOOKS:

- Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
- B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, PHI
- E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
- Yashavant Kanetkar, Let Us C, 18th Edition, BPB
- R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
- Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
- Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
- Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

Sulhan
[Signature]

B.V.K.J

G. Rajeshwari

[Signature]
[Signature]

PYTHON PROGRAMMING LABORATORY**B.Tech. I Year II Sem.**

L	T	P	C
0	1	2	2

Course Objectives:

- To install and run the Python interpreter
- To learn control structures.
- To Understand Lists, Dictionaries in python
- To Handle Strings and Files in Python

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

- Develop the application specific codes using python.
- Understand Strings, Lists, Tuples and Dictionaries in Python
- Verify programs using modular approach, file I/O, Python standard library
- Implement Digital Systems using Python

Note: The lab experiments will be like the following experiment examples

Week -1:

1. i) Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
ii) Start the Python interpreter and type help() to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3. i) Write a program to calculate compound interest when principal, rate and number of periods are given.
ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read name, address, email and phone number of a person through keyboard and print the details.

Week - 2:



1. Print the below triangle using for loop.
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

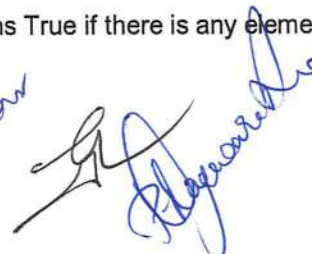
Week - 3:

1. i) Write a program to convert a list and tuple into arrays.
ii) Write a program to find common values between two arrays.
2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.
3. Write a function called palindrome that takes a string argument and returns True if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.

Week - 4:

1. Write a function called is_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.
2. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.

- i). Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
 - ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.
 - iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
 - ii) Remove the given word in all the places in a string?
 - iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5:

1. i) Write a python program that defines a matrix and prints
 - ii) Write a python program to perform addition of two square matrices
 - iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
3. Use the structure of exception handling all general purpose exceptions.

Week-6:

1. a. Write a function called `draw_rectangle` that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
 - b. Add an attribute named `color` to your Rectangle objects and modify `draw_rectangle` so that it uses the `color` attribute as the fill color.
 - c. Write a function called `draw_point` that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
 - d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called `draw_circle` that draws circles on the canvas.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 7

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. Write a function that reads a file `file1` and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.







TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCE BOOKS:

1. Python for Data Science, Dr. Mohd. Abdul Hameed, Wiley Publications - 1st Ed. 2021.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
3. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Think Python, Allen Downey, Green Tea Press
6. Core Python Programming, W. Chun, Pearson
7. Introduction to Python, Kenneth A. Lambert, Cengage

Sethu

Sethu

B. V. S.

G. Rajashellam

R. Jayarama Rao

IT WORKSHOP**B.Tech. I Year II Sem.**

L	T	P	C
0	0	2	1

Course Objectives: The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, PowerPoint and Publisher.

Course Outcomes:

- Perform Hardware troubleshooting
- Understand Hardware components and inter dependencies
- Safeguard computer systems from viruses/worms
- Document/ Presentation preparation
- Perform calculations using spreadsheets

PC Hardware

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 with both Windows and Linux. 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 LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of LaTeX 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 LaTeX







and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

Task 2: Using LaTeX 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 LaTeX 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

Powerpoint

Task 1: Students will be working on basic power point utilities and tools which help them create basic powerpoint 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.

REFERENCE BOOKS:

1. Comdex Information Technology course tool kit Vikas Gupta, *WILEY Dreamtech*
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, *WILEY Dreamtech*
3. Introduction to Information Technology, ITL Education Solutions limited, *Pearson Education*.
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 Third Edition by David Anfinson and Ken Quamme. – *CISCO Press, Pearson Education*.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – *CISCO Press, Pearson Education*.

**ELLENKI COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS), HYDERABAD
PROPOSED SYLLABUS**

**B.Tech. in ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING COURSE STRUCTURE,
I & II YEAR SYLLABUS (ER23 Regulations) Applicable from AY 2023-24**

Batch I YEAR I SEMESTER

S. No.	Course Code	Course	L	T	P	Credits
1.	MA101BS	Matrices and Calculus	3	1	0	4
2.	PH102BS	Applied Physics	3	1	0	4
3.	CS103ES	Programming for Problem Solving	3	0	0	3
4.	ME104ES	Engineering Workshop	0	1	3	2.5
5.	EN105HS	English for Skill Enhancement	2	0	0	2
6.	CS106ES	Elements of Computer Science & Engineering	0	0	2	1
7.	PH107BS	Applied Physics Laboratory	0	0	3	1.5
8.	CS108ES	Programming for Problem Solving Laboratory	0	0	2	1
9.	EN109HS	English Language and Communication Skills Laboratory	0	0	2	1
10.	*MC110	Environmental Science	3	0	0	0
		Induction Program				
Total			14	3	12	20

I YEAR II SEMESTER

S. No.	Course Code	Course	L	T	P	Credits
1.	MA201BS	Ordinary Differential Equations and Vector Calculus	3	1	0	4
2.	CH202BS	Engineering Chemistry	3	1	0	4
3.	ME203ES	Computer Aided Engineering Graphics	1	0	4	3
4.	EE204ES	Basic Electrical Engineering	2	0	0	2
5.	EC205ES	Electronic Devices and Circuits	2	0	0	2
6.	CS206ES	Python Programming Laboratory	0	1	2	2
7.	CH207BS	Engineering Chemistry Laboratory	0	0	2	1
8.	EE208ES	Basic Electrical Engineering Laboratory	0	0	2	1
9.	CS209ES	IT Workshop	0	0	2	1
Total			11	3	12	20

II YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Mathematical and Statistical Foundations	3	1	0	4
2		Data Structures	3	0	0	3
3		Computer Organization and Architecture	3	0	0	3
4		Software Engineering	3	0	0	3
5		Operating Systems	3	0	0	3
6		Introduction to Data Structures Lab	0	0	2	1
7		Operating Systems Lab	0	0	2	1
8		Software Engineering Lab	0	0	2	1
9		Node JS/ React JS/ Django	0	0	2	1
	*MC	Constitution of India	3	0	0	0
Total			18	0	10	20

S. K. Srinivas

S. K. Srinivas

B. V. J.

C. Rajasekhara

S. K. Srinivas

S. K. Srinivas

II YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Discrete Mathematics	3	0	0	3
2		Automata Theory and Compiler Design	3	0	0	3
3		Database Management Systems	3	0	0	3
4		Introduction to Artificial Intelligence	3	0	0	3
5		Object Oriented Programming through Java	3	0	0	3
6		Database Management Systems Lab	0	0	2	1
7		Java Programming Lab	0	0	2	1
8		Real-time Research Project/Field-Based Research Project	0	0	4	2
9		Prolog/ Lisp/ Pyswip	0	0	2	1
10	*MC	Gender Sensitization Lab	0	0	2	1
		Total	15	0	12	20

III YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Design and Analysis of Algorithms	3	1	0	4
2		Machine Learning	3	0	0	3
3		Computer Networks	3	0	0	3
4		Business Economics & Financial Analysis	3	0	0	3
5		Professional Elective-I	3	0	0	3
6		Machine Learning Lab	0	0	2	1
7		Computer Networks Lab	0	0	2	1
8		Advanced English Communication Skills Lab	0	0	2	1
9		UI design- Flutter	0	0	2	1
10	*MC	Intellectual Property Rights	3	0	0	0
		Total	18	01	08	20

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Knowledge Representation and Reasoning	3	0	0	3
2		Data Analytics	3	0	0	3
3		Natural Language Processing	3	0	0	3
4		Professional Elective – II	3	0	0	3
5		Open Elective-I	3	0	0	3
6		Natural Language Processing Lab	0	0	3	1.5
7		Principles of Data Analytics Lab	0	0	3	1.5
8		Industrial Oriented Mini Project/ Internship/Skill Development Course (DevOps)	0	0	4	2
9	*MC	Environmental Science	3	0	0	0
		Total	18	0	10	20

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

IV YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Deep Learning	3	0	0	3
2		Nature Inspired Computing	2	0	0	2
3		Professional Elective -III	3	0	0	3
4		Professional Elective -IV	3	0	0	3
5		Open Elective - II	3	0	0	3
6		Professional Practice, Law & Ethics	0	0	4	2
7		Professional Elective - III Lab	0	0	2	1
8		Project Stage - I	0	0	6	3
		Total Credits	14	0	12	20

IV YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Professional Elective - V	3	0	0	3
2		Professional Elective - VI	3	0	0	3
3		Open Elective - III	3	0	0	3
4		Project Stage - II including Seminar	0	0	22	11
		Total Credits	9	0	22	20

*MC – Satisfactory/Unsatisfactory

#Skill Course - 1 credit with 2 Practical Hours

Professional Elective-I

	Graph Theory
	Introduction to Data Science
	Web Programming
	Image Processing
	Computer Graphics

Professional Elective - II

	Software Testing Methodologies
	Information Retrieval Systems
	Pattern Recognition
	Computer Vision and Robotics
	Data Warehousing and Business Intelligence

Professional Elective - III

	Internet of Things
	Data Mining
	Scripting Languages
	Mobile Application Development
	Cloud Computing

Courses in PE - III and PE - III Lab must be in 1-1 correspondence.






Professional Elective -IV

	Quantum Computing
	Expert Systems
	Semantic Web
	Game Theory
	Mobile Computing

Professional Elective - V

	Social Network Analysis
	Federated Machine Learning
	Augmented Reality & Virtual Reality
	Web Security
	Ad-hoc & Sensor Networks

Professional Elective – VI

	Speech and Video Processing
	Robotic Process Automation
	Randomized Algorithms
	Cognitive Computing
	Conversational AI

Open Elective I:

1. Fundamentals of AI
2. Machine Learning Basics

Open Elective II:

1. Introduction to Natural Language Processing
2. AI applications

Open Elective III:

1. Chatbots
2. Genetic Algorithms & Fuzzy logic

S. Sankar

S. Sankar

B. V. S. R.

A. Rajashekar

S. Sankar

A. Rajashekar

PROGRAMMING FOR PROBLEM SOLVING

B.Tech. I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:

- To learn the fundamentals of computers.
- To understand the various steps in program development.
- To learn the syntax and semantics of the C programming language.
- To learn the usage of structured programming approaches in solving problems.

Course Outcomes: The student will learn

- To write algorithms and to draw flowcharts for solving problems.
- To convert the algorithms/flowcharts to C programs.
- To code and test a given logic in the C programming language.
- To decompose a problem into functions and to develop modular reusable code.
- To use arrays, pointers, strings and structures to write C programs.
- Searching and sorting problems.

UNIT - I: Introduction to Programming

Compilers, compiling and executing a program.

Representation of Algorithm - Algorithms for finding roots of a quadratic equations, finding minimum and maximum numbers of a given set, finding if a number is prime number Flowchart/Pseudocode with examples, Program design and structured programming

Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code, Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators

Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto, Iteration with for, while, do- while loops

I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments

UNIT - II: Arrays, Strings, Structures and Pointers:

Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays

Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings

Structures: Defining structures, initializing structures, unions, Array of structures

Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of Pointers in self-referential structures, usage of self referential structures in linked list (no implementation) Enumeration data type

UNIT - III: Preprocessor and File handling in C:

Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef

Files: Text and Binary files, Creating and Reading and writing text and binary files, Appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

UNIT - IV: Function and Dynamic Memory Allocation:

Functions: Designing structured programs, Declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries

Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions
Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

UNIT - V: Searching and Sorting:

Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms), Basic concept of order of complexity through the example programs

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

ELEMENTS OF COMPUTER SCIENCE AND ENGINEERING

B.Tech. I Year I Sem.

L	T	P	C
0	0	2	1

Course Objective: To provide an overview of the subjects of computer science and engineering.

Course Outcomes:

1. Know the working principles of functional units of a basic Computer
2. Understand program development, the use of data structures and algorithms in problem solving.
3. Know the need and types of operating system, database systems.
4. Understand the significance of networks, internet, WWW and cyber security.
5. Understand Autonomous systems, the application of artificial intelligence.

UNIT – I

Basics of a Computer – Hardware, Software, Generations of computers. Hardware - functional units, Components of CPU, Memory – hierarchy, types of memory, Input and output devices. Software – systems software, application software, packages, frameworks, IDEs.

UNIT – II

Software development – waterfall model, Agile, Types of computer languages – Programming, markup, scripting Program Development – steps in program development, flowcharts, algorithms, data structures – definition, types of data structures

UNIT – III

Operating systems: Functions of operating systems, types of operating systems, Device & Resource management

Database Management Systems: Data models, RDBMS, SQL, Database Transactions, data centers, cloud services

UNIT – IV

Computer Networks: Advantages of computer networks, LAN, WAN, MAN, internet, WiFi, sensor networks, vehicular networks, 5G communication.

World Wide Web – Basics, role of HTML, CSS, XML, Tools for web designing, Social media, Online social networks.

Security – information security, cyber security, cyber laws

UNIT – V

Autonomous Systems: IoT, Robotics, Drones, Artificial Intelligence – Learning, Game Development, natural language processing, image and video processing.

Cloud Basics

TEXT BOOK:

1. Invitation to Computer Science, G. Michael Schneider, Macalester College, Judith L. Gersting University of Hawaii, Hilo, Contributing author: Keith Miller University of Illinois, Springfield.

REFERENCE BOOKS:

1. Fundamentals of Computers, Reema Thareja, Oxford Higher Education, Oxford University Press.
2. Introduction to computers, Peter Norton, 8th Edition, Tata McGraw Hill.
3. Computer Fundamentals, Anita Goel, Pearson Education India, 2010.
4. Elements of computer science, Cengage.

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

PROGRAMMING FOR PROBLEM SOLVING LABORATORY**B.Tech. I Year I Sem.**

L	T	P	C
0	0	2	1

[Note: The programs may be executed using any available Open Source/ Freely available IDE

Some of the Tools available are:

CodeLite: <https://codelite.org/>

Code:Blocks: <http://www.codeblocks.org/>

DevCpp : <http://www.bloodshed.net/devcpp.html>

Eclipse: <http://www.eclipse.org>

This list is not exhaustive and is NOT in any order of preference]

Course Objectives: The students will learn the following:

- To work with an IDE to create, edit, compile, run and debug programs
- To analyze the various steps in program development.
- To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
- To develop modular, reusable and readable C Programs using the concepts like functions, arrays etc.
- To Write programs using the Dynamic Memory Allocation concept.
- To create, read from and write to text and binary files

Course Outcomes: The candidate is expected to be able to:

- formulate the algorithms for simple problems
- translate given algorithms to a working and correct program
- correct syntax errors as reported by the compilers
- identify and correct logical errors encountered during execution
- represent and manipulate data with arrays, strings and structures
- use pointers of different types
- create, read and write to and from simple text and binary files
- modularize the code with functions so that they can be reused

Practice sessions:

- Write a simple program that prints the results of all the operators available in C (including pre/post increment , bitwise and/or/not , etc.). Read required operand values from standard input.
- Write a simple program that converts one given data type to another using auto conversion and casting. Take the values from standard input.

Simple numeric problems:

- Write a program for finding the max and min from the three numbers.
- Write the program for the simple, compound interest.
- Write a program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input.
- Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:
- 5 x 1 = 5
- 5 x 2 = 10
- 5 x 3 = 15
- Write a program that shows the binary equivalent of a given positive number between 0 to 255.

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

Expression Evaluation:

- a. A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula $s = ut + (1/2)at^2$ where u and a are the initial velocity in m/sec ($= 0$) and acceleration in m/sec^2 ($= 9.8 m/s^2$)).
- b. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators $+, -, *, /, \%$ and use Switch Statement)
- c. Write a program that finds if a given number is a prime number
- d. Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
- e. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
- f. Write a C program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
- g. Write a C program to find the roots of a Quadratic equation.
- h. Write a C program to calculate the following, where x is a fractional value.
- i. $1 - x/2 + x^2/4 - x^3/6$
- j. Write a C program to read in two numbers, x and n , and then compute the sum of this geometric progression: $1 + x + x^2 + x^3 + \dots + x^n$. For example: if n is 3 and x is 5, then the program computes $1 + 5 + 25 + 125$.

Arrays, Pointers and Functions:

- a. Write a C program to find the minimum, maximum and average in an array of integers.
- b. Write a function to compute mean, variance, Standard Deviation, sorting of n elements in a single dimension array.
- c. Write a C program that uses functions to perform the following:
- d. Addition of Two Matrices
- e. Multiplication of Two Matrices
- f. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be the same.
- g. Write C programs that use both recursive and non-recursive functions
- h. To find the factorial of a given integer.
- i. To find the GCD (greatest common divisor) of two given integers.
- j. To find x^n
- k. Write a program for reading elements using a pointer into an array and display the values using the array.
- l. Write a program for display values reverse order from an array using a pointer.
- m. Write a program through a pointer variable to sum of n elements from an array.

Files:

- a. Write a C program to display the contents of a file to standard output device.
- b. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
- c. Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.
- d. Write a C program that does the following:
It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function)
Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function)
The program should then read all 10 values and print them back.

Shah

BTU

BTU

Arif Jashwan

BTU

BTU

- e. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

Strings:

- Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
- Write a C program that converts a number ranging from 1 to 50 to Roman equivalent
- Write a C program that uses functions to perform the following operations:
- To insert a sub-string into a given main string from a given position.
- To delete n Characters from a given position in a given string.
- Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)
- Write a C program that displays the position of a character ch in the string S or - 1 if S doesn't contain ch.
- Write a C program to count the lines, words and characters in a given text.

Miscellaneous:

- Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

- Write a C program to construct a pyramid of numbers as follows:

```

1           *           1           1           *
1 2        **          2 3         2 2         **
1 2 3      ***          4 5 6       3 3 3       ***
                                           4 4 4 4      **
                                           *

```

Sorting and Searching:

- Write a C program that uses non recursive function to search for a Key value in a given
- list of integers using linear search method.
- Write a C program that uses non recursive function to search for a Key value in a given
- sorted list of integers using binary search method.
- Write a C program that implements the Bubble sort method to sort a given list of
- integers in ascending order.
- Write a C program that sorts the given array of integers using selection sort in descending order
- Write a C program that sorts the given array of integers using insertion sort in ascending order
- Write a C program that sorts a given array of names

TEXT BOOKS:

- Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
- B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, PHI
- E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
- Yashavant Kanetkar, Let Us C, 18th Edition, BPB
- R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
- Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
- Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
- Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

COMPUTER AIDED ENGINEERING GRAPHICS

B.Tech. I Year II Sem.

L	T	P	C
1	0	4	3

Course Objectives:

- To develop the ability of visualization of different objects through technical drawings
- To acquire computer drafting skill for communication of concepts, ideas in the design of engineering products

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

- Apply computer aided drafting tools to create 2D and 3D objects
- sketch conics and different types of solids
- Appreciate the need of Sectional views of solids and Development of surfaces of solids
- Read and interpret engineering drawings
- Conversion of orthographic projection into isometric view and vice versa manually and by using computer aided drafting

UNIT – I:

Introduction to Engineering Graphics: Principles of Engineering Graphics and their Significance, Scales – Plain & Diagonal, Conic Sections including the Rectangular Hyperbola – General method only. Cycloid, Epicycloid and Hypocycloid, Introduction to Computer aided drafting – views, commands and conics

UNIT- II:

Orthographic Projections: Principles of Orthographic Projections – Conventions – Projections of Points and Lines, Projections of Plane regular geometric figures. Auxiliary Planes. Computer aided orthographic projections – points, lines and planes

UNIT – III:

Projections of Regular Solids – Auxiliary Views - Sections or Sectional views of Right Regular Solids – Prism, Cylinder, Pyramid, Cone – Auxiliary views, Computer aided projections of solids – sectional views

UNIT – IV:

Development of Surfaces of Right Regular Solids – Prism, Cylinder, Pyramid and Cone, Development of surfaces using computer aided drafting

UNIT – V:

Isometric Projections: Principles of Isometric Projection – Isometric Scale – Isometric Views – Conventions – Isometric Views of Lines, Plane Figures, Simple and Compound Solids – Isometric Projection of objects having non-isometric lines. Isometric Projection of Spherical Parts. Conversion of Isometric Views to Orthographic Views and Vice-versa – Conventions. Conversion of orthographic projection into isometric view using computer aided drafting.

TEXT BOOKS:

1. Engineering Drawing N.D. Bhatt / Charotar
2. Engineering Drawing and graphics Using AutoCAD Third Edition, T. Jeyapoovan, Vikas: S. Chand and company Ltd.

REFERENCE BOOKS:

1. Engineering Drawing, Basant Agrawal and C M Agrawal, Third Edition McGraw Hill
2. Engineering Graphics and Design, WILEY, Edition 2020
3. Engineering Drawing, M. B. Shah, B.C. Rane / Pearson.
4. Engineering Drawing, N. S. Parthasarathy and Vela Murali, Oxford
5. Computer Aided Engineering Drawing – K Balaveera Reddy et al – CBS Publishers

Note: - External examination is conducted in conventional mode and internal evaluation to be done by both conventional as well as using computer aided drafting.

(Handwritten signatures and initials in blue and green ink)

PYTHON PROGRAMMING LABORATORY

B.Tech. I Year II Sem.

L	T	P	C
0	1	2	2

Course Objectives:

- To install and run the Python interpreter
- To learn control structures.
- To Understand Lists, Dictionaries in python
- To Handle Strings and Files in Python

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

- Develop the application specific codes using python.
- Understand Strings, Lists, Tuples and Dictionaries in Python
- Verify programs using modular approach, file I/O, Python standard library
- Implement Digital Systems using Python

Note: The lab experiments will be like the following experiment examples

Week -1:

1. i) Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
ii) Start the Python interpreter and type `help()` to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3.
 - i) Write a program to calculate compound interest when principal, rate and number of periods are given.
 - ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read name, address, email and phone number of a person through keyboard and print the details.

Week - 2:

1. Print the below triangle using for loop.

```

5
4 4
3 3 3
2 2 2 2
1 1 1 1 1

```
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

Week - 3:

1. i) Write a program to convert a list and tuple into arrays.
ii) Write a program to find common values between two arrays.
2. Write a function called `gcd` that takes parameters `a` and `b` and returns their greatest common divisor.
3. Write a function called `palindrome` that takes a string argument and returns `True` if it is a palindrome and `False` otherwise. Remember that you can use the built-in function `len` to check the length of a string.

Week - 4:

1. Write a function called `is_sorted` that takes a list as a parameter and returns `True` if the list is sorted in ascending order and `False` otherwise.
2. Write a function called `has_duplicates` that takes a list and returns `True` if there is any element that appears more than once. It should not modify the original list.

Siddhant
Siddhant

B. In
A. Rajesh Kumar

A. Rajesh Kumar

- i). Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
 - ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "l", "a", and the empty string.
 - iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
 - ii) Remove the given word in all the places in a string?
 - iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5:

1. i) Write a python program that defines a matrix and prints
 - ii) Write a python program to perform addition of two square matrices
 - iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
3. Use the structure of exception handling all general purpose exceptions.

Week-6:

1. a. Write a function called `draw_rectangle` that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
 - b. Add an attribute named `color` to your Rectangle objects and modify `draw_rectangle` so that it uses the color attribute as the fill color.
 - c. Write a function called `draw_point` that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
 - d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called `draw_circle` that draws circles on the canvas.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 7

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. Write a function that reads a file `file1` and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.





TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCE BOOKS:

1. Python for Data Science, Dr. Mohd. Abdul Hameed, Wiley Publications - 1st Ed. 2021.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
3. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Think Python, Allen Downey, Green Tea Press
6. Core Python Programming, W. Chun, Pearson
7. Introduction to Python, Kenneth A. Lambert, Cengage

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

IT WORKSHOP**B.Tech. I Year II Sem.**

L	T	P	C
0	0	2	1

Course Objectives: The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, PowerPoint and Publisher.

Course Outcomes:

- Perform Hardware troubleshooting
- Understand Hardware components and inter dependencies
- Safeguard computer systems from viruses/worms
- Document/ Presentation preparation
- Perform calculations using spreadsheets

PC Hardware

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 with both Windows and Linux. 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 LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of LaTeX 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 LaTeX

and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

Task 2: Using LaTeX 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 LaTeX 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

Powerpoint

Task 1: Students will be working on basic power point utilities and tools which help them create basic powerpoint 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.

REFERENCE BOOKS:

1. Comdex Information Technology course tool kit Vikas Gupta, *WILEY Dreamtech*
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, *WILEY Dreamtech*
3. Introduction to Information Technology, ITL Education Solutions limited, *Pearson Education*.
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 Third Edition by David Anfinson and Ken Quamme. – *CISCO Press, Pearson Education*.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – *CISCO Press, Pearson Education*.

**ELLENKI COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS), HYDERABAD
PROPOSED SYLLABUS**

**B.Tech. in COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)
COURSE STRUCTURE, I & II YEAR SYLLABUS (ER23 Regulations)
Applicable from AY 2023-24 Batch**

I Year I Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA101BS	Matrices and Calculus	3	1	0	4
2.	CH102BS	Engineering Chemistry	3	1	0	4
3.	CS103ES	Programming for Problem Solving	3	0	0	3
4.	EE104ES	Basic Electrical Engineering	2	0	0	2
5.	ME105ES	Computer Aided Engineering Graphics	1	0	4	3
6.	CS106ES	Elements of Computer Science & Engineering	0	0	2	1
7.	CH107BS	Engineering Chemistry Laboratory	0	0	2	1
8.	CS108ES	Programming for Problem Solving Laboratory	0	0	2	1
9.	EE109ES	Basic Electrical Engineering Laboratory	0	0	2	1
		Induction Program				
Total			12	2	12	20

I Year II Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA201BS	Ordinary Differential Equations and Vector Calculus	3	1	0	4
2.	PH202BS	Applied Physics	3	1	0	4
3.	ME203ES	Engineering Workshop	0	1	3	2.5
4.	EN204HS	English for Skill Enhancement	2	0	0	2
5.	EC205ES	Electronic Devices and Circuits	2	0	0	2
6.	CS206ES	Python Programming Laboratory	0	1	2	2
7.	PH207BS	Applied Physics Laboratory	0	0	3	1.5
8.	EN208HS	English Language and Communication Skills Laboratory	0	0	2	1
9.	CS209ES	IT Workshop	0	0	2	1
10.	*MC210	Environmental Science	3	0	0	0
Total			13	4	12	20

II YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Digital Electronics	3	0	0	3
2		Data Structures	3	0	0	3
3		Computer Oriented Statistical Methods	3	1	0	4
4		Computer Organization and Architecture	3	0	0	3
5		Object Oriented Programming through Java	3	0	0	3
6		Data Structures Lab	0	0	3	1.5
7		Object Oriented Programming through Java Lab	0	0	3	1.5
8		Data visualization- R Programming/ Power BI	0	0	2	1
9	*MC	Gender Sensitization Lab	0	0	2	0
Total			15	1	10	20

S. Srinivas

B.V.

G. Rajashekhara

G. Srinivas

II YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Discrete Mathematics	3	0	0	3
2		Business Economics & Financial Analysis	3	0	0	3
3		Operating Systems	3	0	0	3
4		Computer Networks	3	0	0	3
5		Software Engineering	3	0	0	3
6		Operating Systems Lab	0	0	2	1
7		Computer Networks Lab	0	0	2	1
8		Real-time Research Project/ Field Based Project	0	0	4	2
9		Node JS/ React JS/ Django	0	0	2	1
10	*MC	Constitution of India	3	0	0	0
Total			18	0	10	20

III YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Network Security and Cryptography	3	1	0	4
2		Database Management Systems	3	0	0	3
3		Formal Languages and Automata Theory	3	0	0	3
4		Professional Elective - I	3	0	0	3
5		Professional Elective - II	3	0	0	3
6		Network Security and Cryptography Lab	0	0	2	1
7		Database Management Systems Lab	0	0	2	1
8		Advanced English Communication Skills Lab	0	0	2	1
9		UI design- Flutter	0	0	2	1
10	*MC	Intellectual Property Rights	3	0	0	0
Total			18	1	8	20

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Cyber Security	3	0	0	3
2		Cyber Crime Investigation & Digital Forensics	3	0	0	3
3		Algorithm Design and Analysis	3	0	0	3
4		Professional Elective – III	3	0	0	3
5		Open Elective - I	3	0	0	3
6		Cyber Security Lab	0	0	2	1
7		Cyber Crime Investigation & Digital Forensics Lab	0	0	2	1
8		Professional Elective – III Lab	0	0	2	1
9		Industrial Oriented Mini Project / Summer Internship/ Skill Development Course (Big data-Spark)	0	0	4	2
10	*MC	Environmental Science	3	0	0	0
Total			18	0	10	20

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

(Handwritten signatures and initials in blue and green ink)

IV YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Vulnerability Assessment & Penetration Testing	3	0	0	3
2		Network Management Systems and Operations	3	0	0	3
3		Professional Elective - IV	3	0	0	3
4		Professional Elective - V	3	0	0	3
5		Open Elective - II	3	0	0	3
6		Vulnerability Assessment & Penetration Testing Lab	0	0	2	1
7		Network Management Systems and Operations Lab	0	0	2	1
8		Project Stage - I	0	0	6	3
Total			15	0	14	20

IV YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Organizational Behavior	3	0	0	3
2		Professional Elective - VI	3	0	0	3
3		Open Elective - III	3	0	0	3
4		Project Stage - II including Seminar	0	0	22	11
Total			9	0	22	20

*MC – Satisfactory/Unsatisfactory

Professional Elective - I

Compiler Design
Artificial Intelligence
Data warehousing and Data Mining
Ad-hoc & Sensor Networks
Cloud Computing

Professional Elective - II

Ethical Hacking
Data Science
Distributed Systems
Cyber Laws
IoT Security

Professional Elective - III

Mobile Application Security
Machine Learning
DevOps
Mobile Application Development
Blockchain Technology

Courses in PE - III and PE - III Lab must be in 1-1 correspondence.

Professional Elective -IV

	Edge Analytics
	Web & Database Security
	Computer Security & Audit Assurance
	Social Media Security
	Deep Learning

Professional Elective -V

	Quantum Computing
	Data Analytics for Fraud Detection
	5G Technologies
	Security Incident & Response Management (SOC)
	Authentication Techniques

Professional Elective – VI

	Quantum Cryptography
	IoT Cloud Processing and Analytics
	Cloud Security
	Digital Watermarking and Steganography
	Data Privacy

Open Elective I:

1. Cyber Laws
2. Ethical Hacking

Open Elective II:

1. Computer Security & Audit Assurance
2. Social Media Security

Open Elective III:

1. Data Privacy
2. 5G Technologies

S. Shankar
2024
B.V.S.
G. Rajashekar
P. Nagaraj

PROGRAMMING FOR PROBLEM SOLVING

B.Tech. I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:

- To learn the fundamentals of computers.
- To understand the various steps in program development.
- To learn the syntax and semantics of the C programming language.
- To learn the usage of structured programming approaches in solving problems.

Course Outcomes: The student will learn

- To write algorithms and to draw flowcharts for solving problems.
- To convert the algorithms/flowcharts to C programs.
- To code and test a given logic in the C programming language.
- To decompose a problem into functions and to develop modular reusable code.
- To use arrays, pointers, strings and structures to write C programs.
- Searching and sorting problems.

UNIT - I: Introduction to Programming

Compilers, compiling and executing a program.

Representation of Algorithm - Algorithms for finding roots of a quadratic equations, finding minimum and maximum numbers of a given set, finding if a number is prime number Flowchart/Pseudocode with examples, Program design and structured programming

Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code, Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators

Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto, Iteration with for, while, do- while loops

I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments

UNIT - II: Arrays, Strings, Structures and Pointers:

Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays

Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings

Structures: Defining structures, initializing structures, unions, Array of structures

Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of Pointers in self-referential structures, usage of self referential structures in linked list (no implementation) Enumeration data type

UNIT - III: Preprocessor and File handling in C:

Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef

Files: Text and Binary files, Creating and Reading and writing text and binary files, Appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

UNIT - IV: Function and Dynamic Memory Allocation:

Functions: Designing structured programs, Declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries

Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions
Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

UNIT - V: Searching and Sorting:

Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms), Basic concept of order of complexity through the example programs

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

S. Hanly
E. B. Koffman

B.A. Forouzan

R.F. Gilberg

McGraw-Hill
Prentice Hall

ELEMENTS OF COMPUTER SCIENCE AND ENGINEERING

B.Tech. I Year I Sem.

L	T	P	C
0	0	2	1

Course Objective: To provide an overview of the subjects of computer science and engineering.

Course Outcomes:

1. Know the working principles of functional units of a basic Computer
2. Understand program development, the use of data structures and algorithms in problem solving.
3. Know the need and types of operating system, database systems.
4. Understand the significance of networks, internet, WWW and cyber security.
5. Understand Autonomous systems, the application of artificial intelligence.

UNIT – I

Basics of a Computer – Hardware, Software, Generations of computers. Hardware - functional units, Components of CPU, Memory – hierarchy, types of memory, Input and output devices. Software – systems software, application software, packages, frameworks, IDEs.

UNIT – II

Software development – waterfall model, Agile, Types of computer languages – Programming, markup, scripting Program Development – steps in program development, flowcharts, algorithms, data structures – definition, types of data structures

UNIT – III

Operating systems: Functions of operating systems, types of operating systems, Device & Resource management

Database Management Systems: Data models, RDBMS, SQL, Database Transactions, data centers, cloud services

UNIT – IV

Computer Networks: Advantages of computer networks, LAN, WAN, MAN, internet, WiFi, sensor networks, vehicular networks, 5G communication.

World Wide Web – Basics, role of HTML, CSS, XML, Tools for web designing, Social media, Online social networks.

Security – information security, cyber security, cyber laws

UNIT – V

Autonomous Systems: IoT, Robotics, Drones, Artificial Intelligence – Learning, Game Development, natural language processing, image and video processing.

Cloud Basics

TEXT BOOK:

1. Invitation to Computer Science, G. Michael Schneider, Macalester College, Judith L. Gersting University of Hawaii, Hilo, Contributing author: Keith Miller University of Illinois, Springfield.

REFERENCE BOOKS:

1. Fundamentals of Computers, Reema Thareja, Oxford Higher Education, Oxford University Press.
2. Introduction to computers, Peter Norton, 8th Edition, Tata McGraw Hill.
3. Computer Fundamentals, Anita Goel, Pearson Education India, 2010.
4. Elements of computer science, Cengage.

PROGRAMMING FOR PROBLEM SOLVING LABORATORY

B.Tech. I Year I Sem.

L	T	P	C
0	0	2	1

[Note: The programs may be executed using any available Open Source/ Freely available IDE

Some of the Tools available are:

CodeLite: <https://codelite.org/>

Code:Blocks: <http://www.codeblocks.org/>

DevCpp : <http://www.bloodshed.net/devcpp.html>

Eclipse: <http://www.eclipse.org>

This list is not exhaustive and is NOT in any order of preference]

Course Objectives: The students will learn the following:

- To work with an IDE to create, edit, compile, run and debug programs
- To analyze the various steps in program development.
- To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
- To develop modular, reusable and readable C Programs using the concepts like functions, arrays etc.
- To Write programs using the Dynamic Memory Allocation concept.
- To create, read from and write to text and binary files

Course Outcomes: The candidate is expected to be able to:

- formulate the algorithms for simple problems
- translate given algorithms to a working and correct program
- correct syntax errors as reported by the compilers
- identify and correct logical errors encountered during execution
- represent and manipulate data with arrays, strings and structures
- use pointers of different types
- create, read and write to and from simple text and binary files
- modularize the code with functions so that they can be reused

Practice sessions:

- a. Write a simple program that prints the results of all the operators available in C (including pre/post increment , bitwise and/or/not , etc.). Read required operand values from standard input.
- b. Write a simple program that converts one given data type to another using auto conversion and casting. Take the values from standard input.

Simple numeric problems:

- a. Write a program for finding the max and min from the three numbers.
- b. Write the program for the simple, compound interest.
- c. Write a program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input.
- d. Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:
- e. $5 \times 1 = 5$
- f. $5 \times 2 = 10$
- g. $5 \times 3 = 15$.
- h. Write a program that shows the binary equivalent of a given positive number between 0 to 255.

S. Anand

B. V. J.

A. Rajasekhara

P. Jayaram

Expression Evaluation:

- A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula $s = ut + (1/2)at^2$ where u and a are the initial velocity in m/sec ($= 0$) and acceleration in m/sec^2 ($= 9.8 m/s^2$)).
- Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators $+, -, *, /, \%$ and use Switch Statement)
- Write a program that finds if a given number is a prime number
- Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
- A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
- Write a C program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
- Write a C program to find the roots of a Quadratic equation.
- Write a C program to calculate the following, where x is a fractional value.
 - $1 - x/2 + x^2/4 - x^3/6$
- Write a C program to read in two numbers, x and n , and then compute the sum of this geometric progression: $1 + x + x^2 + x^3 + \dots + x^n$. For example: if n is 3 and x is 5, then the program computes $1 + 5 + 25 + 125$.

Arrays, Pointers and Functions:

- Write a C program to find the minimum, maximum and average in an array of integers.
- Write a function to compute mean, variance, Standard Deviation, sorting of n elements in a single dimension array.
- Write a C program that uses functions to perform the following:
 - Addition of Two Matrices
 - Multiplication of Two Matrices
 - Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be the same.
- Write C programs that use both recursive and non-recursive functions
- To find the factorial of a given integer.
- To find the GCD (greatest common divisor) of two given integers.
- To find x^n
- Write a program for reading elements using a pointer into an array and display the values using the array.
- Write a program for display values reverse order from an array using a pointer.
- Write a program through a pointer variable to sum of n elements from an array.

Files:

- Write a C program to display the contents of a file to standard output device.
- Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
- Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.
- Write a C program that does the following:
It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using `atoi` function)
Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use `fseek` function)
The program should then read all 10 values and print them back.

- e. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

Strings:

- Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
- Write a C program that converts a number ranging from 1 to 50 to Roman equivalent
- Write a C program that uses functions to perform the following operations:
- To insert a sub-string into a given main string from a given position.
- To delete n Characters from a given position in a given string.
- Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)
- Write a C program that displays the position of a character ch in the string S or - 1 if S doesn't contain ch.
- Write a C program to count the lines, words and characters in a given text.

Miscellaneous:

- a. Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

- b. Write a C program to construct a pyramid of numbers as follows:

```

1           *           1           1           *
1 2        **          2 3         2 2         **
1 2 3      ***          4 5 6       3 3 3       ***
                                     4 4 4       **
                                         *
```

Sorting and Searching:

- Write a C program that uses non recursive function to search for a Key value in a given
- list of integers using linear search method.
- Write a C program that uses non recursive function to search for a Key value in a given
- sorted list of integers using binary search method.
- Write a C program that implements the Bubble sort method to sort a given list of
- integers in ascending order.
- Write a C program that sorts the given array of integers using selection sort in descending order
- Write a C program that sorts the given array of integers using insertion sort in ascending order
- Write a C program that sorts a given array of names

TEXT BOOKS:

- Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
- B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, PHI
- E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
- Yashavant Kanetkar, Let Us C, 18th Edition, BPB
- R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
- Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
- Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
- Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

Handwritten signature

Handwritten signature

Handwritten signature

Handwritten signature

Handwritten signature

PYTHON PROGRAMMING LABORATORY

B.Tech. I Year II Sem.

L	T	P	C
0	1	2	2

Course Objectives:

- To install and run the Python interpreter
- To learn control structures.
- To Understand Lists, Dictionaries in python
- To Handle Strings and Files in Python

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

- Develop the application specific codes using python.
- Understand Strings, Lists, Tuples and Dictionaries in Python
- Verify programs using modular approach, file I/O, Python standard library
- Implement Digital Systems using Python

Note: The lab experiments will be like the following experiment examples

Week -1:

1. i) Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
ii) Start the Python interpreter and type `help()` to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3.
 - i) Write a program to calculate compound interest when principal, rate and number of periods are given.
 - ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read name, address, email and phone number of a person through keyboard and print the details.

Week - 2:

1. Print the below triangle using for loop.

```

5
4 4
3 3 3
2 2 2 2
1 1 1 1 1

```
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

Week - 3:

1. i) Write a program to convert a list and tuple into arrays.
ii) Write a program to find common values between two arrays.
2. Write a function called `gcd` that takes parameters `a` and `b` and returns their greatest common divisor.
3. Write a function called `palindrome` that takes a string argument and returns `True` if it is a palindrome and `False` otherwise. Remember that you can use the built-in function `len` to check the length of a string.

Week - 4:

1. Write a function called `is_sorted` that takes a list as a parameter and returns `True` if the list is sorted in ascending order and `False` otherwise.
2. Write a function called `has_duplicates` that takes a list and returns `True` if there is any element that appears more than once. It should not modify the original list.

Signature
Signature

Signature
A. Rajesh Kumar

Signature
Signature

- i). Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
 - ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.
 - iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
 - ii) Remove the given word in all the places in a string?
 - iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5:

1. i) Write a python program that defines a matrix and prints
 - ii) Write a python program to perform addition of two square matrices
 - iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
 3. Use the structure of exception handling all general purpose exceptions.

Week-6:

1. a. Write a function called `draw_rectangle` that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
 - b. Add an attribute named `color` to your Rectangle objects and modify `draw_rectangle` so that it uses the `color` attribute as the fill color.
 - c. Write a function called `draw_point` that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
 - d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called `draw_circle` that draws circles on the canvas.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
 3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 7

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. Write a function that reads a file `file1` and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

Subhanshu, B.V.J., A. Rajeshwari, and other signatures.

TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCE BOOKS:

1. Python for Data Science, Dr. Mohd. Abdul Hameed, Wiley Publications - 1st Ed. 2021.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
3. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Think Python, Allen Downey, Green Tea Press
6. Core Python Programming, W. Chun, Pearson
7. Introduction to Python, Kenneth A. Lambert, Cengage

S. Ashwin

S. S. S.

B. S. S.

A. Rajashekar

A. Rajashekar

IT WORKSHOP

B.Tech. I Year II Sem.

L T P C
0 0 2 1

Course Objectives: The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, PowerPoint and Publisher.

Course Outcomes:

- Perform Hardware troubleshooting
- Understand Hardware components and inter dependencies
- Safeguard computer systems from viruses/worms
- Document/ Presentation preparation
- Perform calculations using spreadsheets

PC Hardware

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 with both Windows and Linux. 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 LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of LaTeX 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 LaTeX

Handwritten signatures and initials in blue and green ink at the bottom of the page.

and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

Task 2: Using LaTeX 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 LaTeX 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

Powerpoint

Task 1: Students will be working on basic power point utilities and tools which help them create basic powerpoint 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.

REFERENCE BOOKS:

1. Comdex Information Technology course tool kit Vikas Gupta, *WILEY Dreamtech*
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, *WILEY Dreamtech*
3. Introduction to Information Technology, ITL Education Solutions limited, *Pearson Education*.
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 Third Edition by David Anfinson and Ken Quamme. – *CISCO Press, Pearson Education*.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – *CISCO Press, Pearson Education*.

S. Shams

B. S.

G. Rajeshwari

P. Jayaram

ELLENKI COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS), HYDERABAD
PROPOSED SYLLABUS

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)
COURSE STRUCTURE, I & II YEAR SYLLABUS (ER23 Regulations)

Applicable from AY 2023-24 Batch

I Year I Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA101BS	Matrices and Calculus	3	1	0	4
2.	CH102BS	Engineering Chemistry	3	1	0	4
3.	CS103ES	Programming for Problem Solving	3	0	0	3
4.	EE104ES	Basic Electrical Engineering	2	0	0	2
5.	ME105ES	Computer Aided Engineering Graphics	1	0	4	3
6.	CS106ES	Elements of Computer Science & Engineering	0	0	2	1
7.	CH107BS	Engineering Chemistry Laboratory	0	0	2	1
8.	CS108ES	Programming for Problem Solving Laboratory	0	0	2	1
9.	EE109ES	Basic Electrical Engineering Laboratory	0	0	2	1
		Induction Program				
Total			12	2	12	20

I Year II Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA201BS	Ordinary Differential Equations and Vector Calculus	3	1	0	4
2.	PH202BS	Applied Physics	3	1	0	4
3.	ME203ES	Engineering Workshop	0	1	3	2.5
4.	EN204HS	English for Skill Enhancement	2	0	0	2
5.	EC205ES	Electronic Devices and Circuits	2	0	0	2
6.	CS206ES	Python Programming Laboratory	0	1	2	2
7.	PH207BS	Applied Physics Laboratory	0	0	3	1.5
8.	EN208HS	English Language and Communication Skills Laboratory	0	0	2	1
9.	CS209ES	IT Workshop	0	0	2	1
10.	*MC210	Environmental Science	3	0	0	0
Total			13	4	12	20

II YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Digital Electronics	3	0	0	3
2		Data Structures	3	0	0	3
3		Computer Oriented Statistical Methods	3	1	0	4
4		Computer Organization and Architecture	3	0	0	3
5		Object Oriented Programming through Java	3	0	0	3
6		Data Structures Lab	0	0	3	1.5
7		Object Oriented Programming through Java Lab	0	0	3	1.5
8		Data visualization- R Programming/ Power BI	0	0	2	1
9	*MC	Gender Sensitization Lab	0	0	2	0
Total			15	1	10	20

Signature

Signature

Signature

Signature

Signature

II YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Discrete Mathematics	3	0	0	3
2		Business Economics & Financial Analysis	3	0	0	3
3		Operating Systems	3	0	0	3
4		Database Management Systems	3	0	0	3
5		Software Engineering	3	0	0	3
6		Operating Systems Lab	0	0	2	1
7		Database Management Systems Lab	0	0	2	1
8		Real-time Research Project/ Societal Related Project	0	0	4	2
9		Node JS/ React JS/ Django	0	0	2	1
10	*MC	Constitution of India	3	0	0	0
Total			18	0	10	20

III YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Algorithm Design and Analysis	3	0	0	3
2		Data Science	3	1	0	4
3		Computer Networks	3	0	0	3
4		Professional Elective - I	3	0	0	3
5		Professional Elective - II	3	0	0	3
6		R Programming Lab	0	0	2	1
7		Computer Networks Lab	0	0	2	1
8		Advanced English Communication Skills Lab	0	0	2	1
9		ETL-Kafka/Talend	0	0	2	1
10	*MC	Intellectual Property Rights	3	0	0	0
Total			18	1	08	20

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Automata Theory and Compiler Design	3	0	0	3
2		Machine Learning	3	0	0	3
3		Big Data Analytics	3	0	0	3
4		Professional Elective – III	3	0	0	3
5		Open Elective - I	3	0	0	3
6		Machine Learning Lab	0	0	2	1
7		Introduction to Big Data Analytics Lab	0	0	2	1
8		Professional Elective - III Lab	0	0	2	1
9		Industrial Oriented Mini Project/ Summer Internship/ Skill Development Course (UI design- Flutter)	0	0	4	2
10	*MC	Environmental Science	3	0	0	0
Total			18	0	10	20

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

IV YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Predictive Analytics	3	0	0	3
2		Web and Social Media Analytics	3	0	0	3
3		Professional Elective – IV	3	0	0	3
4		Professional Elective – V	3	0	0	3
5		Open Elective – II	3	0	0	3
6		Web and Social Media Analytics Lab	0	0	2	1
7		Predictive Analytics Lab	0	0	2	1
8		Project Stage – I	0	0	6	3
		Total Credits	15	0	10	20

IV YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Organizational Behavior	3	0	0	3
2		Professional Elective – VI	3	0	0	3
3		Open Elective – III	3	0	0	3
4		Project Stage – II including Seminar	0	0	22	11
		Total Credits	9	0	22	20

*MC – Satisfactory/Unsatisfactory

Professional Elective - I

	Data Warehousing and Business Intelligence
	Artificial Intelligence
	Web Programming
	Image Processing
	Computer Graphics

Professional Elective - II

	Spatial and Multimedia Databases
	Information Retrieval Systems
	Software Project Management
	DevOps
	Computer Vision and Robotics

Professional Elective - III

	Software Testing Methodologies
	Data Visualization Techniques
	Scripting Languages
	Mobile Application Development
	Cryptography and Network Security

Courses in PE - III and PE - III Lab must be in 1-1 correspondence.

Professional Elective - IV

	Quantum Computing
	Database Security
	Natural Language Processing
	Information Storage Management
	Internet of Things

Professional Elective - V

	Privacy Preserving in Data Mining
	Cloud Computing
	Data Science Applications
	Mining Massive Datasets
	Exploratory Data Analysis

Professional Elective - VI

	Data Stream Mining
	Web Security
	Video Analytics
	Blockchain Technology
	Parallel and Distributed Computing

Open Elective - I

1. Fundamentals of Data Science
2. R Programming

Open Elective - II

1. Data Mining
2. Data Analytics

Open Elective - III

1. Introduction to social media mining
2. Data Visualization using Python







PROGRAMMING FOR PROBLEM SOLVING

B.Tech. I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:

- To learn the fundamentals of computers.
- To understand the various steps in program development.
- To learn the syntax and semantics of the C programming language.
- To learn the usage of structured programming approaches in solving problems.

Course Outcomes: The student will learn

- To write algorithms and to draw flowcharts for solving problems.
- To convert the algorithms/flowcharts to C programs.
- To code and test a given logic in the C programming language.
- To decompose a problem into functions and to develop modular reusable code.
- To use arrays, pointers, strings and structures to write C programs.
- Searching and sorting problems.

UNIT - I: Introduction to Programming

Compilers, compiling and executing a program.

Representation of Algorithm - Algorithms for finding roots of a quadratic equations, finding minimum and maximum numbers of a given set, finding if a number is prime number Flowchart/Pseudocode with examples, Program design and structured programming

Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code, Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators

Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto, Iteration with for, while, do- while loops

I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments

UNIT - II: Arrays, Strings, Structures and Pointers:

Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays

Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings

Structures: Defining structures, initializing structures, unions, Array of structures

Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of Pointers in self-referential structures, usage of self referential structures in linked list (no implementation) Enumeration data type

UNIT - III: Preprocessor and File handling in C:

Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef

Files: Text and Binary files, Creating and Reading and writing text and binary files, Appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

UNIT - IV: Function and Dynamic Memory Allocation:

Functions: Designing structured programs, Declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries

Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions
Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

UNIT - V: Searching and Sorting:

Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms), Basic concept of order of complexity through the example programs

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

[Handwritten signatures and initials in blue and green ink at the bottom of the page]

ELEMENTS OF COMPUTER SCIENCE AND ENGINEERING**B.Tech. I Year I Sem.**

L	T	P	C
0	0	2	1

Course Objective: To provide an overview of the subjects of computer science and engineering.**Course Outcomes:**

1. Know the working principles of functional units of a basic Computer
2. Understand program development, the use of data structures and algorithms in problem solving.
3. Know the need and types of operating system, database systems.
4. Understand the significance of networks, internet, WWW and cyber security.
5. Understand Autonomous systems, the application of artificial intelligence.

UNIT – I

Basics of a Computer – Hardware, Software, Generations of computers. Hardware - functional units, Components of CPU, Memory – hierarchy, types of memory, Input and output devices. Software – systems software, application software, packages, frameworks, IDEs.

UNIT – II

Software development – waterfall model, Agile, Types of computer languages – Programming, markup, scripting Program Development – steps in program development, flowcharts, algorithms, data structures – definition, types of data structures

UNIT – III

Operating systems: Functions of operating systems, types of operating systems, Device & Resource management

Database Management Systems: Data models, RDBMS, SQL, Database Transactions, data centers, cloud services

UNIT – IV

Computer Networks: Advantages of computer networks, LAN, WAN, MAN, internet, WiFi, sensor networks, vehicular networks, 5G communication.

World Wide Web – Basics, role of HTML, CSS, XML, Tools for web designing, Social media, Online social networks.

Security – information security, cyber security, cyber laws

UNIT – V

Autonomous Systems: IoT, Robotics, Drones, Artificial Intelligence – Learning, Game Development, natural language processing, image and video processing.

Cloud Basics

TEXT BOOK:

1. Invitation to Computer Science, G. Michael Schneider, Macalester College, Judith L. Gersting University of Hawaii, Hilo, Contributing author: Keith Miller University of Illinois, Springfield.

REFERENCE BOOKS:

1. Fundamentals of Computers, Reema Thareja, Oxford Higher Education, Oxford University Press.
2. Introduction to computers, Peter Norton, 8th Edition, Tata McGraw Hill.
3. Computer Fundamentals, Anita Goel, Pearson Education India, 2010.
4. Elements of computer science, Cengage.

Signature

B.Tech

A. Rajesh Kumar

Signature

PROGRAMMING FOR PROBLEM SOLVING LABORATORY

B.Tech. I Year I Sem.

L T P C
0 0 2 1

[Note: The programs may be executed using any available Open Source/ Freely available IDE

Some of the Tools available are:

CodeLite: <https://codelite.org/>

Code:Blocks: <http://www.codeblocks.org/>

DevCpp : <http://www.bloodshed.net/devcpp.html>

Eclipse: <http://www.eclipse.org>

This list is not exhaustive and is NOT in any order of preference]

Course Objectives: The students will learn the following:

- To work with an IDE to create, edit, compile, run and debug programs
- To analyze the various steps in program development.
- To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
- To develop modular, reusable and readable C Programs using the concepts like functions, arrays etc.
- To Write programs using the Dynamic Memory Allocation concept.
- To create, read from and write to text and binary files

Course Outcomes: The candidate is expected to be able to:

- formulate the algorithms for simple problems
- translate given algorithms to a working and correct program
- correct syntax errors as reported by the compilers
- identify and correct logical errors encountered during execution
- represent and manipulate data with arrays, strings and structures
- use pointers of different types
- create, read and write to and from simple text and binary files
- modularize the code with functions so that they can be reused

Practice sessions:

- a. Write a simple program that prints the results of all the operators available in C (including pre/post increment , bitwise and/or/not , etc.). Read required operand values from standard input.
- b. Write a simple program that converts one given data type to another using auto conversion and casting. Take the values from standard input.

Simple numeric problems:

- a. Write a program for finding the max and min from the three numbers.
- b. Write the program for the simple, compound interest.
- c. Write a program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input.
- d. Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:
- e. $5 \times 1 = 5$
- f. $5 \times 2 = 10$
- g. $5 \times 3 = 15$
- h. Write a program that shows the binary equivalent of a given positive number between 0 to 255.

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

[Handwritten signature]

Expression Evaluation:

- A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula $s = ut + (1/2)at^2$ where u and a are the initial velocity in m/sec ($= 0$) and acceleration in m/sec^2 ($= 9.8 m/s^2$)).
- Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators $+$, $-$, $*$, $/$, $\%$ and use Switch Statement)
- Write a program that finds if a given number is a prime number
- Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
- A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
- Write a C program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
- Write a C program to find the roots of a Quadratic equation.
- Write a C program to calculate the following, where x is a fractional value.
- $1 - x/2 + x^2/4 - x^3/6$
- Write a C program to read in two numbers, x and n , and then compute the sum of this geometric progression: $1 + x + x^2 + x^3 + \dots + x^n$. For example: if n is 3 and x is 5, then the program computes $1 + 5 + 25 + 125$.

Arrays, Pointers and Functions:

- Write a C program to find the minimum, maximum and average in an array of integers.
- Write a function to compute mean, variance, Standard Deviation, sorting of n elements in a single dimension array.
- Write a C program that uses functions to perform the following:
- Addition of Two Matrices
- Multiplication of Two Matrices
- Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be the same.
- Write C programs that use both recursive and non-recursive functions
- To find the factorial of a given integer.
- To find the GCD (greatest common divisor) of two given integers.
- To find x^n
- Write a program for reading elements using a pointer into an array and display the values using the array.
- Write a program for display values reverse order from an array using a pointer.
- Write a program through a pointer variable to sum of n elements from an array.

Files:

- Write a C program to display the contents of a file to standard output device.
- Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
- Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.
- Write a C program that does the following:
It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function)
Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function)
The program should then read all 10 values and print them back.

- e. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).

Strings:

- Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
- Write a C program that converts a number ranging from 1 to 50 to Roman equivalent
- Write a C program that uses functions to perform the following operations:
- To insert a sub-string into a given main string from a given position.
- To delete n Characters from a given position in a given string.
- Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)
- Write a C program that displays the position of a character ch in the string S or - 1 if S doesn't contain ch.
- Write a C program to count the lines, words and characters in a given text.

Miscellaneous:

- Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

- Write a C program to construct a pyramid of numbers as follows:

```

1           *           1           1           *
1 2        **          2 3         2 2         **
1 2 3      ***          4 5 6       3 3 3       ***
                                     4 4 4       **
                                     *

```

Sorting and Searching:

- Write a C program that uses non recursive function to search for a Key value in a given list of integers using linear search method.
- Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using binary search method.
- Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.
- Write a C program that sorts the given array of integers using selection sort in descending order
- Write a C program that sorts the given array of integers using insertion sort in ascending order
- Write a C program that sorts a given array of names

TEXT BOOKS:

- Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson
- B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

- Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, PHI
- E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill
- Yashavant Kanetkar, Let Us C, 18th Edition, BPB
- R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)
- Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
- Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition
- Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill







PYTHON PROGRAMMING LABORATORY**B.Tech. I Year II Sem.**

L	T	P	C
0	1	2	2

Course Objectives:

- To install and run the Python interpreter
- To learn control structures.
- To Understand Lists, Dictionaries in python
- To Handle Strings and Files in Python

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

- Develop the application specific codes using python.
- Understand Strings, Lists, Tuples and Dictionaries in Python
- Verify programs using modular approach, file I/O, Python standard library
- Implement Digital Systems using Python

Note: The lab experiments will be like the following experiment examples

Week -1:

1. i) Use a web browser to go to the Python website <http://python.org>. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
ii) Start the Python interpreter and type help() to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3. i) Write a program to calculate compound interest when principal, rate and number of periods are given.
ii) Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read name, address, email and phone number of a person through keyboard and print the details.

Week - 2:

1. Print the below triangle using for loop.
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character (use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

Week - 3:

1. i) Write a program to convert a list and tuple into arrays.
ii) Write a program to find common values between two arrays.
2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.
3. Write a function called palindrome that takes a string argument and returns True if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.

Week - 4:

1. Write a function called is_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.
2. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.

- i). Write a function called `remove_duplicates` that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.
 - ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "l", "a", and the empty string.
 - iii). Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
 - ii) Remove the given word in all the places in a string?
 - iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5:

1. i) Write a python program that defines a matrix and prints
 - ii) Write a python program to perform addition of two square matrices
 - iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
3. Use the structure of exception handling all general purpose exceptions.

Week-6:

1. a. Write a function called `draw_rectangle` that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
 - b. Add an attribute named `color` to your Rectangle objects and modify `draw_rectangle` so that it uses the `color` attribute as the fill color.
 - c. Write a function called `draw_point` that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
 - d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called `draw_circle` that draws circles on the canvas.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 7

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. Write a function that reads a file `file1` and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

Handwritten signature

B. V. S. Reddy

G. Rajeswara Rao

Handwritten signature

TEXT BOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCE BOOKS:

1. Python for Data Science, Dr. Mohd. Abdul Hameed, Wiley Publications - 1st Ed. 2021.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
3. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Think Python, Allen Downey, Green Tea Press
6. Core Python Programming, W. Chun, Pearson
7. Introduction to Python, Kenneth A. Lambert, Cengage

Handwritten signatures and initials in blue and green ink at the bottom of the page.

Signature 1 (Blue): [Illegible]

Signature 2 (Green): B. [Illegible]

Signature 3 (Blue): A. Rajasekhar

Signature 4 (Blue): [Illegible]

IT WORKSHOP

B.Tech. I Year II Sem.

L	T	P	C
0	0	2	1

Course Objectives: The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, PowerPoint and Publisher.

Course Outcomes:

- Perform Hardware troubleshooting
- Understand Hardware components and inter dependencies
- Safeguard computer systems from viruses/worms
- Document/ Presentation preparation
- Perform calculations using spreadsheets

PC Hardware

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 with both Windows and Linux. 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 LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of LaTeX 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 LaTeX

Signature

Signature

Signature

A. Rajagellan

Signature

and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

Task 2: Using LaTeX 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 LaTeX 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

Powerpoint

Task 1: Students will be working on basic power point utilities and tools which help them create basic powerpoint 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.

REFERENCE BOOKS:

1. Comdex Information Technology course tool kit Vikas Gupta, *WILEY Dreamtech*
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, *WILEY Dreamtech*
3. Introduction to Information Technology, ITL Education Solutions limited, *Pearson Education*.
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 Third Edition by David Anfinson and Ken Quamme. – *CISCO Press, Pearson Education*.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – *CISCO Press, Pearson Education*.

**ELLENKI COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS),
HYDERABAD**

PROPOSED SYLLABUS

**M. Tech. COMPUTER SCIENCE AND ENGINEERING/COMPUTER SCIENCE
EFFECTIVE FROM ACADEMIC YEAR 2023 - 24 ADMITTED BATCH**

ER23 COURSE STRUCTURE AND SYLLABUS

I YEAR I – SEMESTER

Course Code	Course Title	L	T	P	Credits
Professional Core - I	Mathematical Foundations of Computer Science	3	0	0	3
Professional Core - II	Advanced Data Structures	3	0	0	3
Professional Elective - I	1. Database Programming with PL/SQL 2. Deep Learning 3. Natural Language Processing	3	0	0	3
Professional Elective - II	1. Applied Cryptography 2. Software Quality Engineering 3. Quantum Computing	3	0	0	3
Lab - I	Advanced Data Structures Lab	0	0	4	2
Lab - II	Professional Elective - I Lab	0	0	4	2
	Research Methodology & IPR	2	0	0	2
Audit - I	Audit Course- I	2	0	0	0
	Total	16	0	8	18

Professional Elective- I and Professional Elective- I Lab must be of same course.

I YEAR II – SEMESTER

Course Code	Course Title	L	T	P	Credits
Professional Core - III	Advanced Algorithms	3	0	0	3
Professional Core - IV	Advanced Computer Architecture	3	0	0	3
Professional Elective - III	1. Enterprise Cloud Concepts 2. Advanced Computer Networks 3. Edge Analytics	3	0	0	3
Professional Elective - IV	1. Bioinformatics 2. Nature Inspired Computing 3. Robotic Process Automation	3	0	0	3
Lab - III	Advanced Algorithms Lab	0	0	4	2
Lab - IV	Professional Elective - III Lab	0	0	4	2
	Mini Project with Seminar	0	0	4	2
Audit - II	Audit Course- II	2	0	0	0
	Total	14	0	12	18

Professional Elective- III and Professional Elective- III Lab must be of same course.

[Handwritten signatures and initials in blue and green ink]

II YEAR III – SEMESTER

Course Code	Course Title	L	T	P	Credits
Professional Elective - V	1. Digital Forensics 2. High Performance Computing 3. Mining Massive Datasets	3	0	0	3
Open Elective	Open Elective	3	0	0	3
Dissertation	Dissertation Work Review - II	0	0	12	6
	Total	6	0	12	12

II YEAR II - SEMESTER

Course Code	Course Title	L	T	P	Credits
Dissertation	Dissertation Work Review - III	0	0	12	6
Dissertation	Dissertation Viva-Voce	0	0	28	14
	Total	0	0	40	20

Note: For Dissertation Work Review - I, Please refer 7.10 in R19 Academic Regulations.

Audit Course I&II:

1. English for Research Paper Writing
2. Disaster Management
3. Sanskrit for Technical Knowledge
4. Value Education
5. Constitution of India
6. Pedagogy Studies
7. Stress Management by yoga
8. Personality Development Through Life Enlightenment Skills

Open Electives for other Departments:

1. IPR
2. Fault Tolerance Systems
3. Intrusion Detection Systems
4. Digital Forensics
5. Optimization Techniques
6. Cyber Physical Systems
7. Graph Analytics

[Handwritten signature]

[Handwritten signature]

C. Rajashellar

[Handwritten signature]

ADVANCED DATA STRUCTURES(PC-II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
3	0	0	3

Prerequisites: A course on "Data Structures"**Course Objectives**

1. Introduces the heap data structures such as leftist trees, binomial heaps, Fibonacci and min-max heaps
2. Introduces a variety of data structures such as disjoint sets, hash tables, search structures and digital search structures

Course Outcomes

1. Ability to select the data structures that efficiently model the information in a problem
2. Ability to understand how the choice of data structures impact the performance of programs
3. Design programs using a variety of data structures, including hash tables, search structures and digital search structures

UNIT - I**Heap Structures**

Introduction, Min-Max Heaps, Leftist trees, Binomial Heaps, Fibonacci heaps.

UNIT - II**Hashing and Collisions**

Introduction, Hash Tables, Hash Functions, different Hash Functions: Division Method, Multiplication Method, Mid-Square Method, Folding Method, Collisions

UNIT - III**Search Structures:** OBST, AVL trees, Red-Black trees, Splay trees,
Multiway Search Trees: B-trees, 2-3 trees**UNIT - IV****Digital Search Structures**

Digital Search trees, Binary tries and Patricia, Multiway Tries, Suffix trees, Standard Tries, Compressed Tries

UNIT - V**Pattern matching**

Introduction, Brute force, the Boyer -Moore algorithm, Knuth-Morris-Pratt algorithm, Naïve String, Harspool, Rabin Karp

TEXT BOOKS:

1. Fundamentals of data structures in C++ Sahni, Horowitz, Mehatha, Universities Press.
2. Introduction to Algorithms, TH Cormen, PHI

REFERENCES:

1. Design methods and analysis of Algorithms, SK Basu, PHI.
2. Data Structures & Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education.
3. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Universities Press.

DATABASE PROGRAMMING WITH PL/SQL (Professional Elective - I)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:

1. Knowledge on significance of SQL fundamentals.
2. Evaluate functions and triggers of PL/SQL
3. Knowledge on control structures, packages in PL/SQL and its applications

Course Outcomes:

1. Understand importance of PL/SQL basics
2. Implement functions and procedures using PL/SQL
3. Understand the importance of triggers in database

UNIT - I

PL/SQL Basics: Block Structure, Behavior of Variables in Blocks, Basic Scalar and Composite Data Types, Control Structures, Exceptions, Bulk Operations, Functions, Procedures, and Packages, Transaction Scope.

UNIT - II

Language Fundamentals & Control Structures: Lexical Units, Variables and Data Types, Conditional Statements, Iterative Statements, Cursor Structures, Bulk Statements, Introduction to Collections, Object Types: Varray and Table Collections, Associative Arrays, Oracle Collection API.

UNIT - III

Functions and Procedures: Function and Procedure Architecture, Transaction Scope, Calling Subroutines, Positional Notation, Named Notation, Mixed Notation, Exclusionary Notation, SQL Call Notation, Functions, Function Model Choices, Creation Options, Pass-by-Value Functions, Pass-by-Reference Functions, Procedures, Pass-by-Value Procedures, Pass-by-Reference Procedures, Supporting Scripts.

UNIT - IV

Packages: Package Architecture, Package Specification, Prototype Features, Serially Reusable Precompiler Directive, Variables, Types, Components: Functions and Procedures, Package Body, Prototype Features, Variables, Types, Components: Functions and Procedures, Definer vs. Invoker Rights Mechanics, Managing Packages in the Database Catalog, Finding, Validating, and Describing Packages, Checking Dependencies, Comparing Validation Methods: Timestamp vs. Signature.

UNIT - V

Triggers: Introduction to Triggers, Database Trigger Architecture, Data Definition Language Triggers, Event Attribute Functions, Building DDL Triggers, Data Manipulation Language Triggers, Statement-Level Triggers, Row-Level Triggers, Compound Triggers, INSTEAD OF Triggers, System and Database Event Triggers, Trigger Restrictions, Maximum Trigger Size, SQL Statements, LONG and LONG RAW Data Types.

TEXT BOOKS:

1. Oracle Database 12c PL/SQL Programming Michael McLaughlin, McGrawHill Education

REFERENCES:

1. Benjamin Rosenzweig, Elena Silvestrova Rakhimov, Oracle PL/SQL by example Fifth Edition
2. Dr. P. S. Deshpande, SQL & PL / SQL for Oracle 11g Black Book

Handwritten signatures and initials at the bottom of the page, including names like "S. S. S.", "B. S.", "A. P. S.", and "P. S. Deshpande".

DEEP LEARNING (Professional Elective - I)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives: students will be able

1. To understand complexity of Deep Learning algorithms and their limitations
2. To be capable of performing experiments in Deep Learning using real-world data.

Course Outcomes:

1. Implement deep learning algorithms, understand neural networks and traverse the layers of data
2. Learn topics such as convolutional neural networks, recurrent neural networks, training deep networks and high-level interfaces
3. Understand applications of Deep Learning to Computer Vision
4. Understand and analyze Applications of Deep Learning to NLP

UNIT - I

Introduction: Feed forward Neural networks, Gradient descent and the back propagation algorithm, Unit saturation, the vanishing gradient problem, and ways to mitigate it. ReLU Heuristics for avoiding bad local minima, Heuristics for faster training, Nestors accelerated gradient descent, Regularization, Dropout

UNIT - II

Convolutional Neural Networks: Architectures, convolution/pooling layers, Recurrent Neural Networks: LSTM, GRU, Encoder Decoder architectures. Deep Unsupervised Learning: Auto encoders, Variational Auto-encoders, Adversarial Generative Networks, Auto-encoder and DBM Attention and memory models; Dynamic Memory Models

UNIT - III

Applications of Deep Learning to Computer Vision: Image segmentation, object detection, automatic image captioning, Image generation with Generative adversarial networks, video to text with LSTM models, Attention Models for computer vision tasks

UNIT - IV

Applications of Deep Learning to NLP: Introduction to NLP and Vector Space Model of Semantics, Word Vector Representations: Continuous Skip-Gram Model, Continuous Bag-of-Wordsmodel (CBOW), Glove, Evaluations and Applications in word similarity

UNIT - V

Analogy reasoning: Named Entity Recognition, Opinion Mining using Recurrent Neural Networks: Parsing and Sentiment Analysis using Recursive Neural Networks: Sentence Classification using Convolutional Neural Networks, Dialogue Generation with LSTMs

TEXT BOOKS:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
2. The Elements of Statistical Learning by T. Hastie, R. Tibshirani, and J. Friedman, Springer.
3. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press.

REFERENCES:

1. Bishop, C, M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.
4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

NATURAL LANGUAGE PROCESSING (Professional Elective - I)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
3	0	0	3

Prerequisites:

1. Data structures, finite automata and probability theory.

Course Objectives:

1. Introduction to some of the problems and solutions of NLP and their relation to linguistics and statistics.

Course Outcomes:

1. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
2. Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems
3. Able to manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
4. Able to design, implement, and analyze NLP algorithms Able to design different language modeling Techniques.
5. Able to design different language modeling Techniques.

UNIT - I

Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models. **Finding the Structure of Documents:** Introduction, Methods, Complexity of the Approaches, Performances of the Approaches

UNIT - II

Syntax Analysis: Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms, Models for Ambiguity Resolution in Parsing, Multilingual Issues.

UNIT - III

Semantic Parsing: Introduction, Semantic Interpretation, System Paradigms, Word Sense Systems, Software.

UNIT - IV

Predicate-Argument Structure, Meaning Representation Systems, Software.

UNIT - V

Discourse Processing: Cohesion, Reference Resolution, Discourse Cohesion and Structure Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation, Types of Language Models, Language-Specific Modeling Problems, Multilingual and Cross Lingual Language Modeling.

TEXT BOOKS:

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary

REFERENCE:

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications

Subhan
 B.V.
 G. Rajeshwari
 P. Nagaraj

APPLIED CRYPTOGRAPHY (Professional Elective - II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives: Knowledge on significance of cryptographic protocols and symmetric and public key algorithms

Course Outcomes:

1. Understand the various cryptographic protocols
2. Analyze key length and algorithm types and modes
3. Illustrate different public key algorithms in cryptosystems
4. Understand special algorithms for protocols and usage in the real world.

UNIT - I

Foundations: Terminology, Steganography, Substitution Ciphers and Transposition Ciphers, Simple XOR, One-Time Pads, Computer Algorithms, Large Numbers,

Cryptographic Protocols: Protocol Building Blocks: Introduction to Protocols, Communications Using Symmetric Cryptography, One-Way Functions, One-Way Hash Functions, Communications Using Public-Key Cryptography, Digital Signatures, Digital Signatures with Encryption, Random and Pseudo-Random-Sequence Generation

UNIT - II

Cryptographic Techniques: Key length: Symmetric Key length, Public key length, comparing symmetric and public key length.

Algorithm types and modes: Electronic Codebook Mode, Block Replay, Cipher Block Chaining Mode, Stream Cipher, Self-Synchronizing Stream Ciphers, Cipher-Feedback Mode, Synchronous Stream Ciphers, Output-Feedback Mod, Counter Mode, Other Block-Cipher Modes.

UNIT - III

Public-Key Algorithms: Background, Knapsack Algorithms, RSA, Pohlig-Hellman, Rabin, ElGamal, McEliece, Elliptic Curve Cryptosystems, LUC, Finite Automaton Public-Key Cryptosystems

Public-Key Digital Signature Algorithms: Digital Signature Algorithm (DSA), DSA Variants, Gost Digital Signature Algorithm, Discrete Logarithm Signature Schemes, Ong-Schnorr-Shamir, ESIGN

UNIT - IV

Special Algorithms for Protocols: Multiple-Key Public-Key Cryptography, Secret-Sharing Algorithms, Subliminal Channel, Undeniable Digital Signatures, Designated Confirmer Signatures, Computing with Encrypted Data, Fair Coin Flips, One-Way Accumulators, All-or-Nothing Disclosure of Secrets, Fair and Failsafe Cryptosystems, Zero-Knowledge Proofs of Knowledge, Blind Signatures, Oblivious Transfer, Secure Multiparty Computation, Probabilistic Encryption, Quantum Cryptography

UNIT - V

Real World Approaches: IBM Secret key management protocol, ISDN, Kerberos, KryptoKnight, Privacy enhanced mail (PEM), Message security protocol (MSP), PGP, Public-Key Cryptography Standards (PKCS), Universal Electronic Payment System (UEPS).

TEXT BOOKS:

1. Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C.(cloth)

Handwritten signatures in blue, green, and black ink at the bottom of the page.

SOFTWARE QUALITY ENGINEERING (Professional Elective - II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives: Knowledge on significance of Quality, quality assurance, quality engineering.

Course Outcomes:

1. Understand software quality and its perspectives
2. Analyze defect prevention and defect reduction in software quality assurance
3. Illustrate software quality engineering activities and its process

UNIT - I

Software Quality: Quality: perspectives and expectations, Quality frameworks and ISO-9126, correctness and defects: Definitions, properties and Measurements, A historical perspective of quality, software quality.

UNIT - II

Quality Assurance: Classification: QA as dealing with defects, Defect prevention- Education and training, Formal method, Other defect prevention techniques, Defect Reduction - Inspection: Direct fault detection and removal, Testing: Failure observation and fault removal, other techniques and risk identification, Defect Containment- software fault tolerance, safety assurance and failure containment

UNIT - III

Quality Engineering: Activities and process, Quality planning: Goal setting and Strategy formation, Quality assessment and Improvement, Quality engineering in software process.

UNIT - IV

Test Activities, Management and Automation: Test planning and preparation, Test execution, Result checking and measurement, Analysis and follow- up, Activities People and Management, Test Automation.

UNIT - V

Coverage and usage testing based on checklist and partitions: Checklist based testing and its limitations, Testing for partition Coverage, Usage based Statistical testing with Musa's operational profiles, Constructing operational profiles

Case Study: OP for the cartridge Support Software

TEXT BOOKS:

1. Jeff Tian, Software Quality Engineering, Testing, Quality Assurance, and Quantifiable improvement
2. Richard N. Taylor, Software Architecture: Foundations, Theory, and Practice

[Signatures: A. Rajasekhar, B. V. S. R. Murthy, P. Jayaram, etc.]

ADVANCED DATA STRUCTURES LAB (Lab - I)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
0	0	4	2

Prerequisites:

1. A course on Computer Programming & Data Structures

Course Objectives:

1. Introduces the basic concepts of Abstract Data Types.
2. Reviews basic data structures such as stacks and queues.
3. Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs, and B-trees.
4. Introduces sorting and pattern matching algorithms.

Course Outcomes:

1. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.
3. Implement and know the application of algorithms for sorting and pattern matching.
4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and B-trees.

List of Programs

1. Write a program to perform the following operations:
 - a) Insert an element into a binary search tree.
 - b) Delete an element from a binary search tree.
 - c) Search for a key element in a binary search tree.
2. Write a program for implementing the following sorting methods:
 - a) Merge sort b) Heap sort c) Quick sort
3. Write a program to perform the following operations:
 - a) Insert an element into a B- tree.
 - b) Delete an element from a B- tree.
 - c) Search for a key element in a B- tree.
4. Write a program to perform the following operations:
 - a) Insert an element into a Min-Max heap
 - b) Delete an element from a Min-Max heap
 - c) Search for a key element in a Min-Max heap
5. Write a program to perform the following operations:
 - a) Insert an element into a Leftist tree
 - b) Delete an element from a Leftist tree
 - c) Search for a key element in a Leftist tree
6. Write a program to perform the following operations:
 - a) Insert an element into a binomial heap
 - b) Delete an element from a binomial heap.
 - c) Search for a key element in a binomial heap
7. Write a program to perform the following operations:
 - a) Insert an element into a AVL tree.
 - b) Delete an element from a AVL search tree.

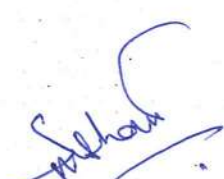
- c) Search for a key element in a AVL search tree.
8. Write a program to perform the following operations:
- Insert an element into a Red-Black tree.
 - Delete an element from a Red-Black tree.
 - Search for a key element in a Red-Black tree.
9. Write a program to implement all the functions of a dictionary using hashing.
10. Write a program for implementing Knuth-Morris-Pratt pattern matching algorithm.
11. Write a program for implementing Brute Force pattern matching algorithm.
12. Write a program for implementing Boyer pattern matching algorithm.

TEXT BOOKS:

- Fundamentals of Data structures in C, E. Horowitz, S. Sahni and Susan Anderson Freed, 2nd Edition, Universities Press
- Data Structures Using C – A.S. Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson education.
- Introduction to Data Structures in C, Ashok Kamthane, 1st Edition, Pearson.

REFERENCES:

- The C Programming Language, B.W. Kernighan, Dennis M. Ritchie, PHI/Pearson Education
- C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press
- Data structures: A Pseudocode Approach with C, R.F. Gilberg And B.A. Forouzan, 2nd Edition, Cengage Learning



DATABASE PROGRAMMING WITH PL/SQL LAB (Lab - II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
0	0	4	2

Course Objectives:

1. Knowledge on significance of SQL fundamentals.
2. Evaluate functions and triggers of PL/SQL
3. Knowledge on control structures, packages in PL/SQL and its applications

Course Outcomes:

1. Understand importance of PL/SQL basics
2. Implement functions and procedures using PL/SQL
3. Understand the importance of triggers in database

List of Experiments:

1. Write a PL/SQL program using FOR loop to insert ten rows into a database table.
2. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID), write a cursor to select the five highest paid employees from the table.
3. Illustrate how you can embed PL/SQL in a high-level host language such as C/Java And demonstrates how a banking debit transaction might be done.
4. Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.
5. Write a PL/SQL program to demonstrate Exceptions.
6. Write a PL/SQL program to demonstrate Cursors.
7. Write a PL/SQL program to demonstrate Functions.
8. Write a PL/SQL program to demonstrate Packages.
9. Write PL/SQL queries to create Procedures.
10. Write PL/SQL queries to create Triggers.

S. Ashwini

B. V. J.

G. Rajakumaran

G. Rajakumaran

DEEP LEARNING LAB (Lab - II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
0	0	4	2

Course Objectives:

1. To Build The Foundation Of Deep Learning.
2. To Understand How To Build The Neural Network.
3. To enable students to develop successful machine learning concepts.

Course Outcomes:

1. Upon the Successful Completion of the Course, the Students would be able to:
2. Learn The Fundamental Principles Of Deep Learning.
3. Identify The Deep Learning Algorithms For Various Types of Learning Tasks in various domains.
4. Implement Deep Learning Algorithms And Solve Real-world problems.

LIST OF EXPERIMENTS:

1. Setting up the Spyder IDE Environment and Executing a Python Program
2. Installing Keras, Tensorflow and Pytorch libraries and making use of them
3. Applying the Convolution Neural Network on computer vision problems
4. Image classification on MNIST dataset (CNN model with Fully connected layer)
5. Applying the Deep Learning Models in the field of Natural Language Processing
6. Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU notes
7. Applying the Autoencoder algorithms for encoding the real-world data
8. Applying Generative Adversial Networks for image generation and unsupervised tasks.

TEXT BOOKS:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
2. The Elements of Statistical Learning by T. Hastie, R. Tibshirani, and J. Friedman, Springer.
3. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press.

REFERENCES:

1. Bishop, C. M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Golub, G.H., and Van Loan C.F., Matrix Computations, JHU Press, 2013.
4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Extensive Reading:

- <http://www.deeplearning.net>
- <https://www.deeplearningbook.org/>
- <https://developers.google.com/machine-learning/crash-course/ml-intro>
- www.cs.toronto.edu/~fritz/absps/imagenet.pdf
- <http://neuralnetworksanddeeplearning.com/>

NATURAL LANGUAGE PROCESSING LAB (Lab - II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
0	0	4	2

Prerequisites: Data structures, finite automata and probability theory**Course Objectives:**

1. To Develop and explore the problems and solutions of NLP.

Course Outcomes:

1. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
2. Able to manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
3. Able to design, implement, and analyze NLP algorithms

List of Experiments

Implement the following using Python

1. Tokenization
2. Stemming
3. Stop word removal (a, the, are)
4. Word Analysis
5. Word Generation
6. Pos tagging
7. Morphology
8. chunking
9. N-Grams
10. N-Grams Smoothing

TEXT BOOKS:

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary.

REFERENCES:

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications.



A. Rajashekar

RESEARCH METHODOLOGY & IPR

M.Tech CSE/CS | Year I Sem.

L	T	P	C
2	0	0	2

Prerequisite: None

Course Objectives:

1. To understand the research problem
2. To know the literature studies, plagiarism and ethics
3. To get the knowledge about technical writing
4. To analyze the nature of intellectual property rights and new developments
5. To know the patent rights

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

1. Understand research problem formulation.
2. Analyze research related information
3. Follow research ethics
4. Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
5. Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to be promoted among students in general & engineering in particular.
6. Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.

UNIT - I:

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations.

UNIT - II:

Effective literature studies approaches, analysis, Plagiarism, Research ethics

UNIT - III:

Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee

UNIT - IV:

Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

UNIT - V:

Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications. New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.







TEXT BOOKS:

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students"
2. C.R. Kothari, Research Methodology, methods & techniques, 2nd edition, New age International publishers

REFERENCES:

1. Ranjit Kumar, 2nd Edition, "Research Methodology: A Step by Step Guide for beginners"
2. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd ,2007.
3. Mayall, "Industrial Design", McGraw Hill, 1992.
4. Niebel, "Product Design", McGraw Hill, 1974.
5. Asimov, "Introduction to Design", Prentice Hall, 1962.
6. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New Technological Age", 2016.
7. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008

Signature

Signature

B.K.

G. Rajesh Kumar

Signature