Regulation: GNITC – R21

COURSE STRUCTURE
&
DETAILED SYLLABUS

for

II Year I Sem
B.Tech. Degree Course

(Applicable for the batch admitted from 2021-22)

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence and Machine Learning)

GNI

GURU NANAK INSTITUTIONS
TECHNICAL CAMPUS
(An Autonomous Institution)
Campus: Ibrahimpatnam, R.R. District - 501506 (T.S).

GNITC is now NAAC A+ Accredited Institute
A+
NAAC

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CONTENTS

CSE II YEAR **II SEM**

S. No.	Subject Code	Group	Subject	L	T	P	Credits
1		ESC	Analog & Digital Electronics	3	0	0	3
2		ESC	Foundations of Computer Organization & Design	3	0	0	3
3		PCC	Data Structures and Algorithms	3	0	0	3
4		PCC	Mathematical Foundations in Computer Science	3	0	0	3
5		PCC	Python Programming for Emerging Technologies	3	0	0	3
6		ESC	Analog & Digital Electronics Lab	0	0	3	1.5
7		PCC	Data Structures and Algorithms Lab	0	0	3	1.5
8		PCC	Python Programming for Emerging Technologies Lab	0	0	3	1.5
9		PCC	IT Workshop Lab	0	0	3	1.5
10		MC	Fundamentals of Artificial Intelligence	2	0	0	0
5 Theory + 4 Lab + 1 MC			**Total Credits**	**17**	**00**	**12**	**21**

II Year B.Tech. Sem-I

L	T	P	C
3	0	0	3

FOUNDATIONS OF COMPUTER ORGANIZATION & DESIGN

PRE-REQUISITE:
1. A course on "Programming for Problem Solving"

CO-REQUISITE:
1. A course on "Analog & Digital Electronics"

COURSE OBJECTIVE:

This course is intended to pertain the knowledge of computer science and electronics engineering to computer hardware and assembly level programming

SYLLABUS:

UNIT – I

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro-operations, shift micro-operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt, Complete Computer Description.

UNIT – II

Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit. Central Processing Unit: General Register Organization, STACK organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT – III

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation, Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT – IV

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access, Input –Output Processor (IOP).

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT – V

Reduced Instruction Set of Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.

TEXT BOOKS:

1. Computer System Architecture – M. Moris Mano, Third Edition, Pearson/PHI.
2. Computer Organization – Car Hamacher, Zvonks Vranesic, SafeaZaky, Vth Edition, McGraw Hill.

REFERENCE BOOKS:

1. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
2. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition PHI/Pearson.

COURSE OUTCOMES:

By the end of the course, the students will be able to

CO 1: Describe the basic structure and fundamentals of computer

CO 2: Discuss the RTL, Micro operations and micro programmed control

CO 3: Interpret the data and storage organization

CO 4: Model the Computer Architectures

CO 5: Write and examine the assembly language programs for various Applications

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (AUTONOMOUS)
SCHOOL OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
(Applicable for the batch admitted from 2021-2022)

II Year B.Tech. Sem-I

L	T	P	C
3	0	0	3

DATA STRUCTURES AND ALGORITHMS

PRE-REQUISITE:

1. A course on "Programming for Problem Solving"

CO-REQUISITE:

1. A course on "Discrete Mathematics"

COURSE OBJECTIVE:

This course provides a comprehensive study of abstract data types, basic data structures, solve the problems using different data structures and design techniques, compare their performance and tradeoffs

SYLLABUS:

UNIT - I

Introduction: Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off.

Searching: Linear Search and Binary Search Techniques and their complexity analysis.

UNIT - II

Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis. ADT queue, Types of Queues: Simple Queue, Circular Queue, Priority Queue; Operations on each type of Queues: Algorithms and their analysis.

UNIT - III

Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: All operations their algorithms and the complexity analysis.

Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees.

UNIT - IV

Sorting and Hashing: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.

UNIT - V

Graph: Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

TEXT BOOKS:

1. "Classic Data Structures", Samanta and Debasis, 2nd edition, PHI publishers.
2. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
3. "Data Structures with C (Schaum's Outline Series)", Seymour Lipschutz, 1st edition, McGraw Hill Education.

REFERENCE BOOKS:

1. Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company.
2. "How to Solve it by Computer", 2nd Impression by R. G. Dromey, Pearson Education.

COURSE OUTCOMES:

By the end of the course, students will be able to

CO 1: Analyze the algorithms to determine the time and computation complexity and justify the correctness;

CO 2: Handle operation like searching, insertion, deletion, traversing on various Data Structures and determine time and computational complexity;

CO 3: Write Sorting algorithms and compare their performance in term of Space and Time complexity

CO 4: Choose appropriate Data Structure as applied to specific problem definition;

CO 5: Demonstrate the reusability of Data Structures for implementing complex iterative problems

II Year B.Tech. Sem-I	L	T	P	C
	3	0	0	3

MATHEMATICAL FOUNDATIONS IN COMPUTER SCIENCE

PRE-REQUISITE:
1. A course on "Mathematics-I"

CO-REQUISITES:
1. A course on "Analog & Digital Electronics"
2. A course on "Data Structures"

COURSE OBJECTIVE:

This course is intended to acquire the ability to work with concepts of discrete structures that includes areas such as functions, relations, sets, predicate logic, combinatorics and graph theory.

SYLLABUS:

UNIT-I

Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Disjunctive and Conjunctive Normal Form, Rules of Inference, The use of Quantifiers.

Proof Techniques: Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency.

UNIT-II

Sets, Relation and Function: Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Sum and Product of Functions, Bijective functions, Inverse and Composite Function, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem.

UNIT-III

Principles of Mathematical Induction: The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic, Basic counting techniques-inclusion and exclusion, pigeon-hole principle, permutation and Combination.

UNIT-IV

Algebraic Structures and Morphism: Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function.

UNIT-V

Graphs and Trees: Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub Graph, Isomorphism, Eulerian and Hamiltonian Walks, Graph Colouring, Colouring maps and Planar Graphs, Colouring Vertices, Colouring Edges, List Colouring, Perfect Graph, definition properties and Example, rooted trees, trees and sorting, weighted trees and prefix codes, Bi-connected component and Articulation Points, Shortest distances.

TEXT BOOKS:

1. J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and It's Application
   to Computer Science, TataMcgraw-Hill
2. C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented
   Approach, 3rd Edition by, Tata McGraw – Hill
3. Abraham Kandel, Joe L. Mott, Theodore P. Baker, Discrete Mathematics for Computer
   Scientists & Mathematicians, 2nd ed. , Pearson Education

REFERENCE BOOKS:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
2. Susanna S. Epp, Discrete Mathematics with Applications,4th edition, Wadsworth Publishing Co. Inc.
3. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press.
4. Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson, Discrete Mathematics, TataMcGraw – Hill
5. Thomas Koshy, Discrete Mathematics with Applications, Elsevier.

COURSE OUTCOMES:

By the end of the course, students will be able to

CO 1: Identify the validity of argument by using propositional and predicate calculus

CO 2: Illustrate the basic terminology of relations, functions and lattices

CO 3: Relate the basic counting techniques to solve the combinatorial problems

CO 4: Produce the recurrence relations through recursively defined structures

CO 5: Apply the basic concepts of graph theory to related theoretical problems

II Year B.Tech. Sem-I

L	T	P	C
3	0	0	3

PYTHON PROGRAMMING

PRE-REQUISITE:

1. A course on "Programming for Problem Solving"

COURSE OBJECTIVE:

This course will enable students to learn Syntax and Semantics and create Functions, handle Strings and Files, understand Lists, Dictionaries and Regular expressions, implement Object Oriented Programming concepts in Python.

SYLLABUS:

UNIT-I

Python Basics, Python Objects- Python Objects, Other Built-in Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types.

Numbers: Introduction to Numbers, Integers, Double Precision Floating Point Real Numbers, Complex Numbers, Operators, Built-in and Factory Functions.

Sequences - Strings, Lists, and Tuples

Mapping and Set Types

Conditionals and Loops

UNIT-II

Files: File Objects, File Built-in Function [open() and file()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution.

Functions and Functional Programming: Calling Functions, Creating Functions, Passing Functions, Formal Arguments, Variable-Length arguments, Functional Programming, Variable Scope, *Recursion, Generators.

Modules: Modules and Files, Namespaces, Importing Modules, Features of Module Import, Module Built-in Functions, Packages

UNIT-III

Regular Expressions: Introduction, Special Symbols and Characters, REs and Python

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

WEB Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application Advanced CGI, Web (HTTP) Servers

UNIT-IV

Database Programming: Introduction, Python Database Application Programmer's Interface (DBAPI), Object Relational Managers (ORMs)

NumPy, SciPy: Introduction, N-dimensional Array in NumPy, Numpy Arrays and Operations, SciPy Basics, Broadcasting in NumPy Array Operations, Array Indexing in NumPy, Constants in NumPy, np.linspace, Understanding np.meshgrid(), Using NumPy, SciPy for Getting Some Basic Information about a Matrix.

Pandas: Open Source Data Analysis and Manipulation Tool: Introduction, Basics of pandas, Using Pandas to Open csv Files

UNIT-V

SymPy: Introduction, The "symbols()" Function, Importing Symbols from Module sympy.abc, Equality Testing in SymPy using "==", Numeric Types, Using Operators on Combination of SymPy Objects and Python Objects, Substitution in a SymPy expression, Convert Python Strings to SymPy Expression and Evaluating it(Functions sympify() and evalf()), Singleton Class in SymPy, Functions in SymPy, Lambda Class in SymPy, Matrices, The linsolve() Method, Differentiation, Integration. Matplotlib, Introduction, An Analogy to Understand Matplotlib, The Figure Class and the Axes Class, Method matplotlib.pyplot.subplot(), Primitives, Creating a Single Plot (no subplots), Format String, Draw sin(x) and cos(x) on Same Plot Using plot(),The Spines and Ticks of a Plot, The plot() Method of pyplot, Draw a Chess-board using Matplotlib, Creating Multiple Subplots, Creating Subplots using subplot() and subplots(), Creating Multiple Figure Objects, Difference between add_axes() and add_subplot(), Creating Scatter Plots, Histograms, Bar-plots and Contour Plots.

TEXT BOOKS:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.
2. Python Programming – Problem Solving, Packages, and Libraries, Anurag Gupta, GP Biswas, McGraw Hill

REFERENCE BOOKS:

1. Think Python, Allen Downey, Green Tea Press
2. Python for Data Science, Mohd.Abdul Hameed, Wiley
3. Introduction to Python, Kenneth A. Lambert, Cengage
4. Core Python Programming, R.Nageswara Rao, Dreamtech Press
5. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
6. Learning Python, Mark Lutz, O'Relly

COURSE OUTCOMES:

By the end of the course, students will be able to

CO 1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions

CO 2: Demonstrate proficiency in handling Strings and File Systems

CO 3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions

CO 4: Interpret the concepts of Object-Oriented Programming as used in Python

CO 5: Implement exemplary applications related to Network Programming, Web Services and Databases in Python

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (AUTONOMOUS)
SCHOOL OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)
(Applicable for the batch admitted from 2021-2022)

II Year B.Tech. Sem-I	L	T	P	C
	0	0	4	2

DATA STRUCTURES AND ALGORITHMS LAB

CO-REQUISITE:

1. A course on "Data Structures"

COURSE OBJECTIVE:

This lab course is intended to write and execute programs in C to solve problems using data structures such as linked lists, stacks, queues, trees, graphs, hash tables search trees, implement various searching and sorting methods

SOFTWARE REQUIREMENTS:

Turbo C / Linux

SYLLABUS:

LIST OF PROGRAMS

1. Write a program to insert a new element at end as well as at a given position in an array.
2. Write a program to find the location of a given element using Linear Search.
3. Write a program to find the location of a given element using Binary Search.
4. Write a program to implement push and pop operations on a stack using linear array.
5. Write a program to convert an infix expression to a postfix expression using stacks.
6. Write a program to evaluate a postfix expression using stacks.
7. Write a program to implement insertion and deletion operations in a queue using linear array.
8. Write a menu driven program to perform following insertion operations in a single linked list:
i. Insertion at beginning ii. Insertion at end iii. Insertion after a given node

iv. Traversing a linked list
9. Write a menu driven program to perform following deletion operations in a single linked list:
i. Deletion at beginning ii. Deletion at end iii. Deletion after a given node

10. Write a program to implement push and pop operations on a stack using linked list.

11. Write a program to implement push and pop operations on a queue using linked list.
12. Program to sort an array of integers in ascending order using bubble sort.
13. Program to sort an array of integers in ascending order using selection sort.
14. Program to sort an array of integers in ascending order using insertion sort.
15. Program to traverse a Binary search tree in Pre-order, In-order and Post-order.
16. Program to traverse graphs using BFS.
17. Program to traverse graphs using DFS.

TEXT BOOKS:
1. "Data Structures with C (Schaum's Outline Series)", Seymour Lipschutz, 1st edition, McGraw Hill Education.

REFERENCE BOOKS:

1. Data structures: A Pseudocode Approach with C, 2nd edition, R.F.Gilberg And B.A.Forouzan, Cengage Learning.
2. Introduction to data structures in c, 1/e Ashok Kamthane.

COURSE OUTCOMES:
The student will be able to:
CO1: Improve practical skills in designing and implementing basic linear data structure algorithms;
CO2: Improve practical skills in designing and implementing Non-linear data structure algorithms;
CO3: Use Linear and Non-Linear data structures to solve relevant problems;
CO4: Choose appropriate Data Structure as applied to specific problem definition; &
CO5: Implement Various searching algorithms and become familiar with their design methods

II Year B.Tech. Sem-I

L	T	P	C
0	0	2	1

PYTHON PROGRAMMINGLAB

CO-REQUISITE:

1. A course on "Python Programming"

COURSE OBJECTIVE:

This lab course is intended to introduce core programming basics and program design with functions using Python programming language and understand the high-performance programs designed to strengthen the practical expertise.

SOFTWARE REQUIREMENTS:

Python

SYLLABUS:

LIST OF PROGRAMS

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers inPython.
3. Write a program to create, concatenate and print a string and accessingsub-string from a given string.
4. Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017"
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a program to program to find largest of three numbers.
9. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]
10. Write a Python program to construct the following pattern, using a nested for loop

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

11. Write a Python script that prints prime numbers less than 20.
12. Write a python program to define a module to find Fibonacci Numbers andimport the module to another program.
13. Write a script named copyfile.py. This script should prompt the user for the names oftwo text files. The contents of the first file should be input and written to the second file.

14. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
15. Write a Python class to convert an integer to a roman numeral.
16. Write a Python class to reverse a string word by word.
17. Write a NumPy program to find the union of two arrays. Union will return the unique, sorted array of values that are in either of the two input array.
18. Write a Pandas program to produce a dataframe object consisting of 5 rows and 4 columns and containing random intergers between 0 and 10.
19. Write a Scipy program to solve the following system of linear equations using the linalg module of Numpy.

$2x+3y+z=13$
$x-y+2x=7$
$3x+4y+z=22$

20. Write a program to create three subplots in the form of 1 x 3 grid, i.e., 1row and 3 columns using Matplotlib.

TEXT BOOKS:

3. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.
4. Python Programming – Problem Solving, Packages, and Libraries, Anurag Gupta, GP Biswas, McGraw Hill

REFERENCE BOOKS:

1. Learning Python, Mark Lutz, O'Relly

COURSE OUTCOMES:

Upon successful completion of this Lab, students will be able to:

CO 1: Write, Test, and Debug Python Programs

CO 2: Implement Conditionals and Loops for Python Programs

CO3: Define and demonstrate the use of built-in data structures lists, tuples and dictionaries.

CO 4: Explore python especially the object-oriented concepts, and the built-in objects of Python

CO 5: Design and implement a program to solve a real-world problem using NumPy, SciPy, SymPy, Pandas, Matplotlib.

II Year B.Tech. Sem-I	L	T	P	C
	0	0	3	1.5

IT WORKSHOP LAB

CO-REQUISITE:

1. A course on “Computer Organization & Architecture”

COURSE OBJECTIVE:

This lab course is introduced to make the students learn about PC Hardware, install operating systems, Hardware and Software Troubleshooting, student how to use Internet and World Wide Web and to use the productivity tools

HARDWARE / SOFTWARE REQUIREMENTS: A typical PC / MS Office / LaTeX

SYLLABUS:

LIST OF TASKS:

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: Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva

Task 4: Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Internet & World Wide Web

Task 5: 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. Customize browsers.

Task 6: Develop your home page using HTML Consisting of your photo, name, address and education details as a table and your skill set as a list.

Productivity Tools

LaTeX and Word

Task 7: 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 8 : 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

Task 9: Excel Orientation: The mentor needs to tell the importance of MS office 2007/ equivalent (FOSS) tool Excel as a Spreadsheet tool, give the details of the two tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources. Creating a Scheduler - Features to be covered:- Gridlines, Format Cells, Summation, auto fill, Formatting Text, calculating GPA

Power Point

Task 10: Students making their presentations interactive. Topic covered during this week includes: Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts

TEXT BOOKS:

1. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
2. Latex Companion –Leslie Lamport, PHI/Pearson.

REFERENCE BOOKS:

1. Introduction to Computers, Peter Norton, 6/e Mc Graw Hill Publishers.
2. Upgrading and Repairing, PC's, 18th e, Scott Muller QUE, Pearson Education.
3. Comdex Information Technology course tool kit Vikas Gupt, WILEY Dreamtech.
4. IT Essentials PC Hardware and Softwre Companion Guide, Third Edition by David Anfinson and ken Quamme. –CISCO Press, Pearson Education.
5. PC Hardware and A+ Handbook—Kate J. Chase PHI (Microsoft)
6. Basic Concepts of Information Technology Workshop,3e, G Prabeen Babu, M V Narayana BS Publications.

COURSE OUTCOMES:

Upon successful completion of this Lab, students will be able to:

CO 1: Attain knowledge for computer assembling and software installation

CO 2: Analyze the hardware and software trouble shooting problems

CO 3: Effectively utilize internet and work on world wide web

CO 4: Apply the tools for personal and professional utilities

CO 5: Create the documents, presentations, and spreadsheets

II Year B.Tech. I Sem

L	T	P	C
3	0	0	0

FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE

PRE-REQUISITES: NIL

COURSE OBJECTIVE:
The course is introduced to familiarize the basic concepts of artificial intelligence, its relevance in the modern era and various applications

SYLLABUS:

UNIT- I
Introduction – What is artificial intelligence, foundations of artificial intelligence, history of artificial intelligence
Intelligent Agents – agents and environments, the structure, good behaviour: the concept of rationality, the nature of environments, the structure of agents, applications of AI.

UNIT - II
Solving Problem by Searching – Problem solving agents, example problems, searching for solutions.
Uninformed Search Strategies – Breadth first search, uniform-cost search, depth first search, depth limited search, iterative deepening search, bidirectional search, comparing uninformed search strategies.

UNIT - III
Logical Agents – Knowledge-based agents, the wumpus world, logic, propositional logic: the very simple logic.
Knowledge Representation - Introduction, approaches to knowledge representation-relational knowledge, knowledge represented as logic, procedural knowledge, knowledge representation using semantic networks, inheritance in semantic net.

UNIT - IV
Expert System & Applications – Introduction, phases in building expert systems-knowledge engineering, knowledge representation, expert systems architecture-knowledgebase, inference engine, knowledge acquisition, expert systems versus traditional systems-characteristics of expert systems, evolution of expert systems, advantages and disadvantages of expert systems, languages for expert system development., applications of expert systems.

UNIT - V

Machine Learning Paradigm – Introduction, machine learning system-components of learning system, rote learning, learning by taking advice.

Supervised & Unsupervised Learning – supervised concept learning, unsupervised concept learning, reinforcement learning.

TEXT BOOKS:

1. Artificial Intelligence-A Modern Approach, 3rd Edition, Stuart J. Russell, Peter Norvig, Pearson Education.
2. Artificial Intelligence, Saroj Kaushik, Cengage Publication

REFERENCE BOOKS:

1. Artificial Intelligence, Elaine Rich, Kevin Knight, Shivashankar B. Nair, 3rd Edition, McGraw Hill.
2. Principles of Artificial Intelligence, Nils J. Nilson, Morgan Kaufmann Publishers.
3. Artificial Intelligence, 3rd Edition, Patric Henry Winston, Pearson Education.
4. Artificial Intelligence Illuminated, Ben Coppin, Narosa Publication

COURSE OUTCOMES:

By the end of the course, students will be able to

CO 1: identify the importance of artificial intelligence

CO 2: apply various search strategies to provide efficient solutions for problem space

CO 3: comprehend various approaches for knowledge representation

CO 4: employ expert systems for knowledge engineering applications

CO 5: develop models using machine learning techniques