SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

SreeSainath Nagar, Tirupati

Department of Computer Science and Engineering

Supporting Document for 1.1.2

Syllabus Revision carried out in 2020

Program: B.Tech.-Computer Science and Engineering

Regulations: SVEC-20

This document details the following:

1. Courses where syllabus has been changed 20% and more.
2. Course-wise revised syllabus with changes highlighted.

Note:*For SVEC-20 revised syllabus, SVEC-19 (previous syllabus) is the reference.*

List of Courses where syllabus content has been changed (20% and more)

S. No.	Course Code	Name of the course	Percentage of Content changed	Page Number in which Details are Highlighted
1.	20BT30503	Python Programming	33.3	2
2.	20BT30532	Python ProgrammingLab	59	6
3.	20BT30533	Workshop in Computer Science	37.5	11
4.	20BT30534	Programming through C++ Lab	100	13
5.	20BT31231	Software Engineering Lab	20	16
6.	20BT40305	Human Resource Management	100	22
7.	20BT50505	Software Project Management	100	24
8.	20BT41531	Design and Analysis of Algorithms Lab	100	26
9.	20BT50532	Python for Data Science Lab	100	28
10.	20BT5MC01	Professional Ethics	100	31
11.	20BT70501	Computer Vision	100	33
12.	20BT71206	Game Development	100	35
13.	20BT70506	Service Oriented Architecture and Microservices Architecture	100	37
14.	20BT70507	Nature Inspired Algorithms	100	39
15.	20BT70531	Computer Vision Lab	100	41
16.	20BT70532	R Programming Lab	100	43
Average % **(A)**			84.36	-
Total No. of Courses in the Program **(T)**			124	
No. of Courses where syllabus (more than 20% content) has been changed**(N)**			16	
Percentage of syllabus content change in the courses **(C) = (A x N) /100**			13.49	
Percentage of Syllabus Content changed in the Program (P) = C/T*100			**10.88**	

DEAN (Academics)

DEAN (Academic)
SREE VIDYANIKETHAN ENGINEERING COLLEGE
Sree Sainath Nagar, A. RANGAMPET
CHITTOOR (DT.)-517 102, A.P.

PRINCIPAL

PRINCIPAL
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)
Sree Sainath Nagar, A. RANGAMPET
Chittoor (Dist.) - 517 102, A.P., INDIA.

II B. Tech. – I Semester

(20BT30503) PYTHON PROGRAMMING

(Common to CSE, CSE(AI) and CSE(DS))

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	3	-	-	3

PRE-REQUISITES: A Course on "Object Oriented Programming through Java"

COURSE DESCRIPTION: Basics of Python programming, Control structures, Lists, Tuples, Strings, Sets, Dictionaries, Regular expressions, Functions, File handling, Object-oriented programming, Exception handling.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Demonstrate knowledge on Python constructs, sequences, sets and dictionaries to solve basic computational problems.

CO2. Apply the concepts of regular expressions for searching patterns in strings.

CO3. Develop and use Python modules to provide solutions to problems.

CO4. Apply the knowledge of file operations in Python for file processing.

CO5. Design applications using object-oriented programming features – encapsulation, inheritance, polymorphism and exception handling.

DETAILED SYLLABUS:

UNIT–I: INTRODUCTION TO PYTHON PROGRAMMING (7 periods)

Introduction to Python, Tokens, Variables, Literals, Identifiers, Keywords, Special symbols, Operators, Fundamental datatypes, Expressions, Type conversions, Handling Input and output in Python.

UNIT–II: CONTROL STRUCTURES (8 periods)

Selection Statements: if statement, if-else statement, if-elif-else statement, nested-if statement.

Iterative Statements: while loop, for loop, break statement, continue statement, pass and else statements used with loops.

UNIT–III: SEQUENCES, SETS, DICTIONARIES AND REGULAR EXPRESSIONS (11 periods)

Sequences: Lists and operations – Creating, Inserting elements, Updating elements, Deleting elements, Searching and sorting, List comprehensions, Nested lists; Tuples – Creating, Searching and sorting, Nested tuples; Strings – Initializing a string and string operations, String handling methods, String formatting.

Sets: Set creation, Set operations.

Dictionaries: Operations on dictionaries, Dictionary methods, Sorting elements using lambdas.

Regular Expressions: Regular expressions, Sequence characters in regular expressions, Quantifiers in regular expressions, Special characters in regular expressions.

UNIT–IV: FUNCTIONS AND FILE HANDLING **(9 periods)**

Functions: Need for functions, Function definition, Function call, Variable scope and lifetime, Return statement, Positional arguments, Keyword arguments, Default arguments and variable length arguments, Recursive functions, Lambda functions, Generators.

File Handling: Types of files, Opening and closing files, Reading and writing data.

UNIT–V: OBJECT ORIENTED PROGRAMMING AND EXCEPTION HANDLING **(10 periods)**

Object Oriented Programming: Introduction to object-oriented programming, Classes and objects, Inheritance and polymorphism, Abstract Classes and interfaces.

Exception Handling: Errors in a python program, Exceptions, Exception handling, Types of exceptions, Except block, Assert statement, User defined exceptions.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOK:

1. R. Nageswara Rao, *Core Python Programming*, 2nd Edition, Dreamtech Press, 2018.

REFERENCE BOOKS:

1. Reema Thareja, *Python Programming using Problem Solving Approach*, 1st Edition, Oxford University Press, 2017.
2. Charles Dierbach, *Introduction to Computer Science using Python: A Computational Problem Solving Focus*, Wiley India, 2016.

ADDITIONAL LEARNING RESOURCES:

- https://onlinecourses.nptel.ac.in/noc19_cs41/preview
- https://www.coursera.org/specializations/python
- https://www.coursera.org/learn/python-for-applied-data-science-ai

I B. Tech. – I Semester

(19BT10501) PROGRAMMING FOR PROBLEM SOLVING

(Common to CE, ME, CSE, CSSE and IT)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	3	1	-	4

PRE-REQUISITES: A course on Basic Mathematics

COURSE DESCRIPTION: Introduction to problem solving approach, Introduction to Python programming, control structures, sequences, sets, Dictionaries, Implementation of Data structures using Python, Modular programming, file handling, Data representation and Visualization.

COURSE OUTCOMES: After successful completion of the course, students will be able to:

CO1. Demonstrate knowledge on Python constructs to solve basic problems.

CO2. Develop and use Python modules to provide solutions to problems.

DETAILED SYLLABUS:

UNIT-I: INTRODUCTION TO PROBLEM SOLVING AND PYTHON PROGRAMMING (10 Periods)

Problem Solving Aspect: top-down design, implementation of algorithms, building blocks of flow charts, program verification and efficiency of algorithms.

Python Programming: tokens, literals, identifiers, keywords, special symbols and operators; fundamental data types, expressions, type conversions, handling Input and output in Python.

UNIT-II: CONTROL STRUCTURES (8 Periods)

Selection Statements: if statement, if-else statement, if-elif-else statement, nested-if statement.

Iterative Statements: while loop, for loop, break statement, continue statement, pass and else statements used with loops.

UNIT-III: SEQUENCES, SETS, DICTIONARIES AND DATA STRUCTURES (9 Periods)

Sequences: Lists and operations - creating, inserting elements, updating elements, deleting elements, searching and sorting, list comprehensions, nested lists; **tuples -** creating, searching and sorting, nested tuples; **strings -** Initializing a string and string operations, string handling methods, string formatting; **sets -** set creation and operations; **dictionaries -** operations on dictionaries, dictionary methods, sorting elements using lambdas.

Data structures: Stacks - push, pop, peek and display operations on stack, applications of stack; **Queues –** enqueue, dequeue and display operations on queue, applications of queues.

UNIT-IV: MODULAR PROGRAMMING AND FILE HANDLING (10 Periods)

Modular Programming: need for functions, function definition, function call, variable scope and lifetime, return statement, positional arguments, keyword arguments, default arguments and variable-length arguments, recursive functions; Modules - math, NumPy, date and time.

File Handling: types of files, opening and closing files, reading and writing data.

UNIT-V: DATA REPRESENTATION AND VISUALIZATION (8 Periods)

Pandas: creating data frame, reading data from CSV files, indexing and selecting data, dealing with rows and columns; Visualization - bar plots, histogram, Scatter Plot.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOKS:

1. R. Nageswara Rao, *Core Python Programming,* 2nd edition, Dreamtech Press, 2018.
2. R. G. Dromey, *How to solve it by Computer,* Pearson, 2006.

REFERENCE BOOKS:

1. Reema Thareja, *Python Programming using Problem Solving Approach*, 1st edition, Oxford University Press, 2017.
2. Charles Dierbach, *Introduction to Computer Science using Python: A Computational Problem-Solving Focus*, Wiley India, 2016.

II B. Tech. – I Semester

(20BT30532) PYTHON PROGRAMMING LAB

(Common to CSE, CSE(AI) and CSE(DS))

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	-	3	1.5

PRE-REQUISITES: A Course on "Python Programming"

COURSE DESCRIPTION: Hands on practice on python programming for problem solving, Sequences, Sets, Dictionaries, Regular expressions, Functions, File handling, Object-oriented concepts through Python, Exception handlings.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Demonstrate practical knowledge on Python constructs, sequences, sets and dictionaries to solve basic computational problems.

CO2. Apply the concepts of regular expressions for searching patterns in strings.

CO3. Develop and use Python modules to provide solutions to problems.

CO4. Apply the knowledge of file operations in Python for file processing.

CO5. Design applications using object-oriented programming features – encapsulation, inheritance, polymorphism and exception handling.

CO6. Work independently to solve problems with effective communication.

LIST OF EXERCISES:

1) a) Write a python script to read two integer numbers and perform arithmetic operations.

   b) Write a python script to evaluate following expressions by considering necessary inputs.
   i) $ax^2 + bx + c$ ii) $ax^5 + bx^3 + c$ iii) $(ax + b) / (ax - b)$ iv) $x - a / b + c$

2) a) Write a python script to convert a given decimal number into octal, hexadecimal and binary.

   b) Write a python script to read four integer values separated with commas and display the sum of those four numbers.

   c) Write a python script to print "SVEC" with prefix of ten spaces by using format().

3) a) Write a python script to calculate electricity bill based on following slab rates.

Consumption units	Rate (in Rupees/Unit)
0-100	4
101-150	4.6
151-200	5.2
201-300	6.3
Above 300	8

(Hint: To get Consumption units take current meter reading and old meter reading from the user as input)

b) Print the following pattern using python script.

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

4) a) Write a python script to check whether the given number (N) can be expressed as Power of Two (2) or not. For example, 8 can be expressed as 2^3.

b) Write a python script to perform following sets operations.
i) update () ii) discard() iii) issuperset() iv) isdisjoint()
v) symmetric_difference ()

5) a) Given an integer tuple, for each element in the tuple check whether there exists a smaller element on the next immediate position of the tuple. If it exists print the smaller element. If there is no smaller element on the immediate next to the element then print -1.
Example: Input: 4 2 1 5 3 Output: 2 1 -1 3 -1

b) Write a Python program to print the number of times the string "SVEC" can be formed for the given input string.
Example: Input: acdfksekfevskdjfs Output: 1

6) a) Write a python script to read details of N students – name, roll number, branch and age. Sort the student details based on their names and display.

b) Write a python script to delete duplicate strings from a list of strings. (Insertion order should be maintained after deleting duplicate string).

c) Write a python script to read details of N students into nested list and convert it into a nested dictionary.

7) a) Design a function that can perform sum of two or three or four numbers.

b) Write a python script to implement Towers of Hanoi problem.

c) Write a python script to print the numbers that do not appear in the Fibonacci series of n numbers where n is given by the user. (Hint: If n is 8 then up to 8 Fibonacci numbers has to be printed Ex: 1 1 2 3 5 8 13 21 and in this series missing numbers should be traced and printed, Ex: missing numbers are: 4 6 7 9 10 11 12 14 15 16 17 18 19.

8) a) Write a python script to create a regular expression to extract all words with 5 characters length from a given string.

b) Write a python script to create a regular expression to extract the phone number from a given text.

9) a) Write a python script to copy the content of one file into another file.

b) Write a python script to read all the strings from the text file and display them.

10) a) Write a Python program for the following problem: Create a base class Basic_Info with data members name, rollno, gender and two member functions getdata() and display(). Derive a class Physical_Fit from Basic_Info which has data members height and weight and member functions getdata() and display(). Display all the information using object of derived class.

b) Define a class REPORT with the following specification:
Private members:
Adno: 4-digit admission number

Name:20 characters
Marks: A list of 5 floating point values
Average: average marks obtained
GETAVG () a function to compute the average obtained in five subjects.
Public members:
READINFO () function to accept values for adno, name, marks. Invoke the function GETAVG ().
DISPLAYINFO () function to display all data members of report on the screen. You should give function definitions. Write driver code to demonstrate all the functions.

11) a) Write a python script to handle following set of exceptions:
i) ArithmeticError ii) ImportError iii) IndexErro iv) KeyError

b) Write a python script to throw a user defined exception called Negative, if the entered input is a negative number.

REFERENCE BOOKS:

1. R. Nageswara Rao, *Core Python Programming*, 2nd Edition, Dreamtech Press, 2018.
2. Reema Thareja, *Python Programming using Problem Solving Approach*, 1st Edition, Oxford University Press, 2017.
3. Charles Dierbach, *Introduction to Computer Science using Python: A Computational Problem Solving Focus*, Wiley India, 2016.

ADDITIONAL LEARNING RESOURCES:

- https://onlinecourses.nptel.ac.in/noc19_cs41/preview
- https://www.coursera.org/specializations/python
- https://www.coursera.org/learn/python-for-applied-data-science-ai

I B. Tech. – I Semester

(19BT10531) PROGRAMMING FOR PROBLEM SOLVING LAB

(Common to CE, ME, CSE, CSSE and IT)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	2	1

PRE-REQUISITES: A course on Basic Mathematics

COURSE DESCRIPTION: The course is designed to provide hands on practice on Scratch programming and python programming for problem solving.

COURSE OUTCOMES: After successful completion of the course, students will be able to:

CO1. Develop scripts using Scratch tool to simulate simple problems.

CO2. Apply Python Constructs and Modules to develop solutions for real-life problems.

CO3. Function effectively as an individual and in team to foster knowledge and creativity.

CO4. Write and present a substantial technical report/ document effectively.

PRACTICAL EXERCISES:

1) a) Design a script in Scratch to simulate Airplane for take-off and land.

   b) Design a script in Scratch to make a sprite to ask the user to enter two different numbers and an arithmetic operator and then calculate and display the result.

2) a) Design a script in Scratch to calculate factorial of a given number.

   b) Design a script in Scratch to simulate Maze game. (Hint: To get Maze images refer http://inventwithScratch.com/downloads/)

3) a) Write a python script to read two integer numbers and perform arithmetic operations.

   b) Write a python script to evaluate following expressions by considering necessary inputs.

   i) $ax^2 + bx + c$ ii) $ax^5 + bx^3 + c$ iii) $(ax + b) / (ax - b)$ iv) $x - a / b + c$

4) a) Write a python script to convert given decimal number into octal, hexa decimal and binary.

   b) Write a python script to read four integer values separated with commas and display the sum of those four numbers.

   c) Write a python script to print “SVEC” with prefix of ten spaces by using format().

5) a) Write a python script to calculate electricity bill based on following slab rates.

Consumption units	Rate (in Rupees/Unit)
0-100	4
101-150	4.6
151-200	5.2
201-300	6.3
Above 300	8

(Hint: To get Consumption units take current Meter reading, old meter reading from the user as input)

b) Print the following pattern using python script.

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

6) a) Write a python script to read *N* student details like name, roll number, branch and age. Sort the student details based on their names and display.

b) Write a python script to delete duplicate strings from a list of strings. (Insertion order should maintain after deleting duplicate string).

c) Write a python script to read N number of student details into nested list and convert that as a nested dictionary.

7) a) Design a function that can perform sum of two or three or four numbers.

b) Write a python script to implement towers of Hanoi problem.

c) Write a Python function primesquare(l) that takes a nonempty list of integers and returns True if the elements of l alternate between perfect squares and prime numbers, and returns False otherwise. Note that the alternating sequence of squares and primes may begin with a square or with a prime. Here are some examples to show how your function should work.

```
>>>primesquare([4])
   True
   >>>primesquare([4,5,16,101,64])
   True
   >>>primesquare([5,16,101,36,27])
   False
```

8) a) Write a python script to perform arithmetic operations on numpyarrays.

b) Write a python script to perform following matrix operations using numpy.

i) Dot product ii) Matrix product iii) Determinant iv) Inverse

9) a) Write a python script to Create Pandas dataframe using list of lists.

b) Write a python script to load data from a CSV file into a Pandas DataFrame and perform basic operations on it.

10) a) Draw a Scatter Plot by considering an appropriate data set.

b) Draw histograms by considering an appropriate data set.

11) Mini Project-1

12) Mini Project-2

TEXT BOOK:

1. R. Nageswara Rao, *Core Python Programming*, 2nd edition, Dreamtech Press, 2018.

II B. Tech. – I Semester

(20BT30533) WORKSHOP IN COMPUTER SCIENCE

(Common to CSE, CSE(AI) and CSE(DS))

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	-	3	1.5

PRE-REQUISITES: A Course on "Python Programming"

COURSE DESCRIPTION: Hands on practice sessions on MS Office, Cloud productivity and collaboration tools, AI Tools, GitHub.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Use office suite for effective documentation, data analysis and presentations.

CO2. Use Cloud-based tools for effective collaboration and data sharing.

CO3. Use AI tools for speech recognition, language detection and object detection.

CO4. Use online code hosting platforms such as GitHub for hosting and collaborating software projects.

CO5. Work independently to solve problems with effective communication.

LIST OF EXERCISES:

MS-Office:

1. MS Word
   a) Introduction to MS-Word, Importance of Word as Word Processor, Overview of toolbars, Saving, Accessing files, Using help and resources. Create a word document using the features: Formatting fonts, Drop cap, Applying text effects, Using character spacing, Borders and shading, Inserting headers and footers, Using date and time option.

   b) Create a word document in MS-Word using the features: Inserting tables, Bullets and numbering, changing text direction, Hyperlink, Images from files and Clipart, Drawing toolbar and Word art and create an invitation using Mail Merge in MS-Word.

2. MS Power Point
   a) Introduction to MS-Power Point, Utilities, Overview of toolbars, PPT orientation, slide layouts, Types of views. Create a Power Point Presentation using the features: Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows.

   b) Create a Power Point Presentation using the features: Auto content wizard, Hyperlinks, Inserting images, Clip art, Audio, Video, Custom animation, Slide hiding, Tables and Charts.

3. MS Excel
   a) Introduction to MS-Excel as a Spreadsheet tool, Overview of toolbars, accessing, Saving excel files, Using help and resources. Create a spreadsheet using the features: Gridlines, Format cells, Summation, Auto fill, Formatting text, Formulae in Excel Charts.

b) Create a spreadsheet using the features: Split cells, Sorting, Conditional formatting, Freeze panes, Pivot tables, Data validation.

Cloud Based Productivity and Collaboration Tools:

4. Introduction to Google drive, Google Backup & Sync, Google Forms.

   a) Store, sync, and share files in the cloud using Google Drive, Google Backup & Sync tools.

   b) Create Google forms to manage event registrations/surveys/quizzes and analyze responses.

AI Tools:

5. Speech Recognition: Read an audio file with Python and use the Google speech recognition API to perform conversion of Speech to Text.

6. Language Detection: Detect the language of the text using language detection library (langdetect) ported from Google's language-detection.

7. Object Detection: Detect multiple objects present in an image using Detectron-Facebook's API.

GitHub:

8. a) Introduction to GitHub basic concepts and Flow.

   b) GitHub Installation and Setup.

   c) Create a “Hello World” project on GitHub by performing the following operations

      - Create and use a repository
      - Create and manage a new branch
      - Perform and commit changes to a file
      - Open and merge pull requests

SOFTWARE/TOOLS USED:

- MS Office
- Google Cloud based productivity and collaboration tools
- Google speech recognition API, Google's language-detection API, Detectron-Facebook's API
- Github

ADDITIONAL LEARNING RESOURCES:

- https://gsuite.google.com/learning-center/products/#!/
- https://pypi.org/project/langdetect/
- https://cloud.google.com/translate/docs
- https://ai.facebook.com/tools/detectron/

II B. Tech. – I Semester

(20BT30534) PROGRAMMING THROUGH C++ LAB

(Skill Oriented Course)

(Common to CSE, CSE(AI) and CSE(DS))

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	1	2	2

PRE-REQUISITES: A Course on "Programming for Problem Solving"

COURSE DESCRIPTION: Hands-on experience on programming through C++, Control statements, Arrays, Strings, Functions, Classes and objects, Polymorphism, Inheritance, Exception Handling, Standard Template Library.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Apply syntactic constructs of C++ programming language to solve logic-based problems.

CO2. Develop application programs using object-oriented programming features.

CO3. Design general algorithms using generic programming through Standard Template Library.

CO4. Work independently to solve problems with effective communication.

LIST OF EXERCISES:

Object-Oriented Programming concepts, Fundamental Datatypes, Constants, Variables, Input and Output Streams, Operators, Control Flow statements, Arrays, Functions, Strings.

1. a) Write a C++ program to perform basic arithmetic operations.

   b) Given an integer variable "days" that contains a number of days, write a C++ program that displays

) the number of whole weeks corresponding to days. For example, if days contains 23, then the number of whole weeks is 3.

      i) the number of days remaining after taking the whole weeks out of the value in days. For example, if days contains 23, then the number of days remaining after 3 whole weeks is 2.

2. a) Write a C++ program that determines the median of three input numbers. The median is the middle number when the three numbers are arranged in order by size. However, the user can input the values in any order.

   b) Write an interactive C++ program which takes input of a series of n temperatures entered by the user, calculates he average temperature.

3. a) ROT13 (rotate by 13 places) is a simple letter substitution cipher that is an instance of a Caesar cipher developed in ancient Rome and used by Julius Caesar who used it in his private correspondence. ROT13 replaces a letter with the letter 13 letters after it in the alphabet. The following example demonstrates the translation in ROT13:

   A → N B → O C→P D → Q

   Thus, the translation of the word JULIUS using ROT13 would be WHYVHF.

Write a C++ program that inputs a sentence and outputs the corresponding translated sentence using the ROT13 encoding scheme above.

b) Write a C++ program that displays a checkerboard pattern made of stars and blanks, as shown below. A checkerboard is of size eight squares by eight squares.

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

4. a) Write a C++ program that reads time in numeric form and displays it in English. The time is input as hours and minutes, separated by a space. Hours are specified in 24-hour time (15 is 3 P.M.), but the output should be in 12-hour A.M./P.M. form. Note that noon and midnight are special cases. Here are some examples:

Input1: Enter Time: 12 00   Output1: Noon
Input2: Enter time: 0 00   Output2: Midnight
Input3: Enter time: 6 44   Output3: Six forty four AM
Input4: Enter time: 18 11   Output4: Six eleven PM

b) Beautiful String - Claudia is playing with strings. She has two strings and she wanted both of them to be beautiful. According to Claudia, both the strings are beautiful if one of the strings is a permutation of the other. Write a C++ program to determine whether the two given strings are beautiful or not.

Input Format: contains two string S and P.

Output Format: Print "YES" (without quotes) if the given strings are beautiful else print "NO" (without quotes).

Examples:
Input1: hello olelh   Output1: YES
Input2: jgec cgec   Output2: NO

5. a) An array A of N integers is given where every element appears twice except for two elements, which appears exactly one time. Write a C++ program to find those numbers which appear only once.

Input Format: First line of input contains a single integer N - denoting number the integers. Second line contains N space separated integers.

Output Format: Print the two elements which appear exactly once in increasing order.

Example: Input: 6
1 2 1 3 3 5
Output: 2 5

b) Write a C++ program to reverse the given input string by keeping its special characters at the same place.

Example: Input: intell#ect   Output: tcelle#tni

Classes and objects, Inheritance, Virtual Functions, Polymorphism, Exceptions and Overloading Operators.

6. a) Write a C++ program for the following problem:
Create a base class Basic_Info with data members name, rollno, gender and two member functions getdata() and display(). Derive a class Physical_Fit from Basic_Info which has data members height and weight and member

functions getdata() and display(). Display all the information using object of derived class.

b) Write a C++ program to overload binary + operator.

c) Write a C++ program for the following problem: Create three classes Person, Professor and Student. The class Person should have data members – name and age, pure virtual functions – getdata() and putdata(). The classes Professor and Student should inherit from the class Person. The class Professor should have integer member: num_publications. There will be two member functions: getdata() and putdata(). The function getdata() should get the input from the user: name, age and number of publications of the professor. The function putdata() should print the name, age, publications of the professor. The class Student should have data member: marks. It has two member functions: getdata() and putdata(). The function getdata() should get the input from the user: name, age, and the marks of the student. The function putdata() should print the name, age, marks of the student. Create one object each for the classes Professor and Student, read and display the data of a professor and a student.

7. a) Write a C++ program to monitor network access points in a building. Each building has a specified number of floors, a certain number of rooms per floor, and some number of network access points in a room. Each network access point is labeled with its state as being on or off and if it is on it has the month it was turned on. Design, implement, and test classes that represent buildings, floors, and network access points. The classes should provide suitable observer methods that allow to determine how many floors are in a building, how many rooms are on a floor, how many access points are in a room, and if an access point is on or off and which month it was turned on. Also provide mutator methods that allow to turn a network access point on or off and to set the month if it is turned on. Use composition as a key strategy in designing these classes—thus, a building should contain floors, floors should contain rooms, and rooms should contain network access points.

b) Design, implement, and test a class in C++ that represents an amount of time in minutes and seconds. The class should provide a constructor that sets the time to a specified number of minutes and seconds. The default constructor should create an object for a time of zero minutes and zero seconds. The class should provide observers that return the minutes and the seconds separately, and an observer that returns the total time in seconds (minutes × 60 + seconds). Boolean comparison observers should be provided that test whether two times are equal, one time is greater than the other, or one time is less than the other. A function should be provided that adds one time to another, and another function that subtracts one time from another. The class should not allow negative times (subtraction of more time than is currently stored should result in a time of 0:00).

Template Classes, Generic Functions, Standard Template Library (STL) - vector, stack, priority_queue, map, bitsets and iterators.

8. a) Write a function using templates that returns the sum of all the elements in a one-dimensional array. The array elements can be of any simple numeric type, and the function has two parameters: the base address of the array and the number of elements in the array.

b) Write a C++ program to implement linear search algorithm. Use vector STL to store N integers.

II B.Tech. – II Semester

(20BT31231) SOFTWARE ENGINEERING LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	-	3	1.5

PRE-REQUISITES: Courses on "Software Engineering" and "Object Oriented Programming through Java"

COURSE DESCRIPTION: Software development life cycle activities- requirements specification, SRS preparation, Modeling case studies –Online Ticket Reservation system; Point of sales.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Analyze user requirements and prepare software requirements specifications.

CO2. Apply design principles of UML for software design.

CO3. Apply tools for developing UML diagrams.

CO4. Use cost estimation models for project evaluation.

CO5. Work independently or in teams to solve problems with effective communication.

LIST OF CASE STUDIES:

Case studies given below should be Modeled using Visual Modeling tools in different views i.e., Use case view, logical view, component view, Deployment view.

CASE STUDY 1: ONLINE TICKET RESERVATION FOR RAILWAYS

Problem Statement: Computer play an integral part of the day in today's life. It makes the entire job easier and faster, every job is computerized so as the ticket reservation we can book over the online ticket reservation system. During the booking of the ticket reservation passenger has to select origin, date of journey, destination, class of train etc. The reservation counter keeps track of passenger's information. Thus the system will have all the details about the trains and facilities provided by them. There are various trains with the different level of convenience for the passengers. The whole database will be maintained by database administrator. There are varieties of trains where the passengers can select the train according to the convenience for their destination journey. The journey could be within the state or across the India. Each train has the three types of classes i.e. Sleeper class, First class and the AC compartment. Design the application for the above problem description.

CASE STUDY 2: A POINT OF SALE (POS) SYSTEM

Problem Statement: A POS System is a computerized application used to record sales and handle payments; it is typically used in a retail store. It includes hardware components such as a computer and bar code scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services and temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client - side terminals and interfaces such as browser, PDA's, touch - screens.

CASE STUDY 3: RECRUITMENT PROCEDURE FOR SOFTWARE INDUSTRY

Problem Statement: In the software industry the recruitment procedure is the basic thing that goes in the hand with the requirement as specified by the technical management team. HR first gives an advertisement in leading Newspapers, Journals, Weeklies and Websites. The job seekers can apply for it through by Post or by e-mail to the company. The technical skill and the experience of the candidates are reviewed and the short listed candidates are called for the interview. There may be different rounds for interview like the written test, technical interview, and HR interview. After the successful completion of all rounds of interview, the selected candidates' names are displayed. Meanwhile HR gives all the details about the salary, working hours, terms and conditions and the retirement benefit to the candidate.

CASE STUDY 4: ONLINE AUCTION SALES

Problem Statement: The online auction system is a design about a website where sellers collect and prepare a list of items they want to sell and place it on the website for visualizing. To accomplish this purpose the user has to access the site. Incase it's a new user he has to register. Purchaser's login and select items they want to buy and keep bidding for it. Interacting with the purchasers and sellers through messages does this. There is no need for customer to interact with the sellers because every time the purchasers bid, the details will be updated in the database. The purchaser making the highest bid for an item before the close of the auction is declared as the owner of the item. If the auctioneer or the purchaser doesn't want to bid for the product then there is fixed cutoff price mentioned for every product. He can pay that amount directly and own the product. The purchaser gets a confirmation of his purchase as an acknowledgement from the website. After the transition by going back to the main menu where he can view other items.

CASE STUDY 5: TWO FLOOR ELEVATOR SIMULATOR

Problem Statement: The elevator has the basic function that all elevator systems have, such as moving up and down, open and close doors, and of course, pick up passengers. The elevator is supposed to be used in a building having floors numbered from 1 to MaxFloor, where the first floor is the lobby. There are car call buttons in the car corresponding to each floor. For every floor except for the top floor and the lobby, there are two hall call buttons for the passengers to call for going up and down. There is only one down hall call button at the top floor and one up hall call button in the lobby. When the car stops at a floor, the doors are opened and the car lantern indicating the current direction the car is going is illuminated so that the passengers can get to know the current moving direction of the car. The car moves fast between floors, but it should be able to slow down early enough to stop at a desired floor. When an elevator has no requests, it remains at its current floor with its doors closed.

In order to certificate system safety, emergency brake will be triggered and the car will be forced to stop under any unsafe conditions.

CASE STUDY 6: HOME APPLIANCE CONTROL SYSTEM

Problem Statement: A home appliance control system (HACS) is a system which provides various services to remotely operate on home appliances, such as microwave oven, TV, and garage door etc through remote devices such as mobile phone, desktop and palm-top. A home appliance control system (HACS) is a system which is controlled by a remote system such as a mobile phone or a palm-top, and at the same time controls, monitors and coordinates home appliances such as air conditioner, microwave oven, garage doors, TV set, VCR, audio controller, indoor/outdoor lights, water sprinkler, home security system, bath tub controller, etc. In order to activate home appliances and to allow for different ways of cooking, the HACS needs mechanisms for communication between the different devices in the system, and for coordination among the various processes running on such devices. The system administrator of the HACS system has the ability to add a new appliance or delete an existing one. The system administrator has the ability to add a new remote device and configure it with HACS or delete an existing one when it is not used. Also, the system administrator can create an account for a new user or delete existing account if it is no longer used.

1. Identify Functional and Non Functional Requirements for

   i) Online Ticket Reservation for Railways, ii) Online Auction Sales

2. Prepare the Software Requirement Specification (SRS), High Level Design (HLD) and Detailed Design (DD) for the following experiments

   i) Employee Information System, ii) Online Airline Reservation

   Note: For the reference of SRS, HLD and DD templates refer department manual and use any open source Requirement documentation tool. Estimate project parameters such as size, effort and time for development for a Library Information system using Basic COCOMO model.

3. Mini project: For any given problem identify functional and non- functional requirements, prepare SRS document and design the system using UML diagrams.

REFERENCE BOOKS:

1. Grady Booch, James Rum Baugh and Ivar Jacobson, *The Unified Modeling Language User Guide*, Pearson Education, Second Edition, 2009.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons and David Fado, *UML 2 Toolkit*, WILEY-Dreamtech India Pvt. Ltd., 2003.
3. Rajesh Naik and Swapna Kishore, *Software Requirements and Estimation*, Tata McGraw Hill, New Delhi, 2001.

SOFTWARE/TOOLS USED:

- Visual Paradigm for modeling diagrams

II B.Tech. – II Semester

(19BT31232) SOFTWARE ENGINEERING LAB

(Common to CSE and CSSE)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	2	1

PRE-REQUISITES: A course on "Software Engineering"

COURSE DESCRIPTION: Software Development Life Cycle activities-requirements specification, SRS preparation, Modeling case studies–Online Ticket Reservation system; Point of sales.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Analyse user requirements and prepare software requirements specifications.

CO2. Apply design principles of UML for software design.

CO3. Apply tools for developing UML diagrams.

CO4. Use cost estimation models for project evaluation.

CO5. Work effectively as an individual to design UML models.

CO6. Write and present a substantial technical report/document effectively.

LIST OF EXPERIMENTS:

1. Identify Functional and Non-Functional Requirements for:
   i) Online Ticket Reservation for Railways ii) Online Auction Sales

2.a) Construct a flow graph for Insertion sort algorithm.

b) Write a program to find Cyclomatic complexity for the above flow graph
(**Hint**: McCabe's cyclomatic matrices V(G) of a graph G with n vertices , e edges and P connected components is V(G)=e-n+2P)

CASE STUDIES:

Case studies given below should be Modeled using Visual Modeling tools in different views i.e. Use case view, logical view, component view, Deployment view.

CASE STUDY 1: ONLINE TICKET RESERVATION FOR RAILWAYS

Problem Statement: Computer play an integral part of the day in today's life. It makes the entire job easier and faster, every job is computerized so as the ticket reservation we can book over the online ticket reservation system. During the booking of the ticket reservation passenger has to select origin, date of journey, destination, class of train etc. The reservation counter keeps track of passenger's information. Thus the system will have all the details about the trains and facilities provided by them. There are various trains with the different level of convenience for the passengers. The whole database will be maintained by database administrator. There are varieties of trains where the passengers can select the train according to the convenience for their destination journey. The journey could be within the state or across the India. Each train has the three types of classes i.e. Sleeper class, First class and the AC compartment. Design the application for the above problem description.

CASE STUDY 2: A POINT OF SALE (POS) SYSTEM

Problem Statement: A POS System is a computerized application used to record sales and handle payments; it is typically used in a retail store. It includes hardware components such as a computer and bar code scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services and temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client – side terminals and interfaces such as browser, PDA's, touch – screens.

CASE STUDY 3: RECRUITMENT PROCEDURE FOR SOFTWARE INDUSTRY

Problem Statement: In the software industry the recruitment procedure is the basic thing that goes in the hand with the requirement as specified by the technical management team. HR first gives an advertisement in leading Newspapers, Journals, Weeklies and Websites. The job seekers can apply for it through by Post or by e-mail to the company. The technical skill and the experience of the candidates are reviewed and the short listed candidates are called for the interview. There may be different rounds for interview like the written test, technical interview, and HR interview. After the successful completion of all rounds of interview, the selected candidates' names are displayed. Meanwhile HR gives all the details about the salary, working hours, terms and conditions and the retirement benefit to the candidate.

CASE STUDY 4: ONLINE AUCTION SALES

Problem Statement: The online auction system is a design about a website where sellers collect and prepare a list of items they want to sell and place it on the website for visualizing. To accomplish this purpose the user has to access the site. Incase it's a new user he has to register. Purchaser's login and select items they want to buy and keep bidding for it. Interacting with the purchasers and sellers through messages does this. There is no need for customer to interact with the sellers because every time the purchasers bid, the details will be updated in the database. The purchaser making the highest bid for an item before the close of the auction is declared as the owner of the item. If the auctioneer or the purchaser doesn't want to bid for the product then there is fixed cutoff price mentioned for every product. He can pay that amount directly and own the product. The purchaser gets a confirmation of his purchase as an acknowledgement from the website. After the transition by going back to the main menu where he can view other items.

CASE STUDY 5: TWO FLOOR ELEVATOR SIMULATOR

Problem Statement: The elevator has the basic function that all elevator systems have, such as moving up and down, open and close doors, and of course, pick up passengers. The elevator is supposed to be used in a building having floors numbered from 1 to Max Floor, where the first floor is the lobby. There are car call buttons in the car corresponding to each floor. For every floor except for the top floor and the lobby, there are two hall call buttons for the passengers to call for going up and down. There is only one down hall call button at the top floor and one up hall call button in the lobby. When the car stops at a floor, the doors are opened and the car lantern indicating the current direction the car is going is illuminated so that the passengers can get to know the current moving direction of the car. The car moves fast between floors, but it should be able to slow down early enough to stop at a desired floor. When an elevator has no requests, it remains at its current floor with its doors closed.

In order to certificate system safety, emergency brake will be triggered and the car will be forced to stop under any unsafe conditions.

CASE STUDY 6: HOME APPLIANCE CONTROL SYSTEM

Problem Statement: A home appliance control system (HACS) is a system which provides various services to remotely operate on home appliances, such as microwave oven, TV, and garage door etc through remote devices such as mobile phone, desktop and palm-top. A home appliance control system (HACS) is a system which is controlled by a remote system such as a mobile phone or a palm-top, and at the same time controls, monitors and coordinates home appliances such as air conditioner, microwave oven, garage doors, TV set, VCR, audio controller, indoor/outdoor lights, water sprinkler, home security system, bath tub controller, etc. In order to activate home appliances and to allow for different ways of cooking, the HACS needs mechanisms for communication between the different devices in the system, and for coordination among the various processes running on such devices. The system administrator of the HACS system has the ability to add a new appliance or delete an existing one. The system administrator has the ability to add a new remote device and configure it with HACS or delete an existing one when it is not used. Also the system administrator can create an account for a new user or delete existing account if it is no longer used.

REFERENCE BOOKS:

1. Grady Booch, James Rum Baugh and Ivar Jacobson, "The Unified Modeling Language User Guide," Second Edition, Pearson Education, 2009.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons and David Fado, "UML 2 Toolkit," WILEY-Dreamtech India Pvt. Ltd., 2003.
3. Rajesh Naik and Swapna Kishore, "Software Requirements and Estimation," Tata McGraw Hill, New Delhi, 2001.

SOFTWARE/TOOLS USED:

Visual Paradigm for modeling diagrams

III B.Tech. – I Semester

(20BT40305) HUMAN RESOURCE MANAGEMENT

(Open Elective - 1)

(Common to CSE, CSE(AI), CSE(DS), CSBS, CSSE and IT)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	3	-	-	3

PRE-REQUISITES: --

COURSE DESCRIPTION: Introduction to HRM; recruitment, training and Placement, talent management; Basic types of interviews; Components; Compensation and reward administration; Ethics, Employee Relations, and Fair Treatment at Work, Dispute resolution and grievance management, trade unions and their role in collective bargaining;

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Demonstrate the knowledge of concepts and perspective on HRM.

CO2. Demonstrate the knowledge of recruitment, placement and talent management.

CO3. Demonstrate the knowledge of training and development.

CO4. Demonstrate the knowledge of compensation and reward administration.

CO5. Demonstrate the knowledge of ethics, and employee relations.

DETAILED SYLLABUS:

UNIT-I: INTRODUCTION TO HUMAN RESOURCE MANAGEMENT (9 Periods)

Importance of Human Resource Management (HRM), Concepts and perspective on HRM, Role of HRM in a competitive business environment, Environment affecting HRM at the enterprise level, The trends shaping HRM, HR metrics and benchmarking, HR accounting, inventory and information systems.

UNIT-II: RECRUITMENT, PLACEMENT, AND TALENT MANAGEMENT (9 Periods)

HR planning and job analysis, Employee testing and selection, Basic types of interviews, Placement, induction and socialization of the employees, Talent acquisition, talent management.

UNIT-III: TRAINING AND DEVELOPMENT (9 Periods)

Employee Orientation/Onboarding, Appraising employee performance and potential evaluation, Techniques for appraising performance, Managing Employee Retention, Engagement, and Careers, Human Resource Development (HRD).

UNIT-IV: COMPENSATION (8 Periods)

Compensation and reward administration, Basic factors in determining pay rates, Job evaluation methods, Pay for Performance and Financial Incentives, Employee benefits and services.

UNIT-V: ENRICHMENT TOPICS IN HRM **(10 Periods)**

Ethics, Employee Relations, and Fair Treatment at Work, Dispute resolution and grievance management, trade unions and their role in collective bargaining, Employee Safety and Health, Global Challenges in HRM, Managing Human Resources in Small and Entrepreneurial Firms, Business environment and workforce adjustments, Creating high performance systems, Innovations in HRM.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOKS:

1. Garry Dessler and Biju Varkkey, *Human Resource Management*, Pearson education, 16th Edition, 2020.
2. Aswathappa K, *Human Resource Management*, McGraw Hill Education, 8th Edition, 2017.

REFERENCE BOOKS:

1. V.S.P Rao, *Human Resource Management*, Taxmann Publications Pvt. Ltd, 2nd Edition 2020.
2. Bohlander George W, Snell Scott, *Principles of Human Resource Management*, Cengage Learning, 16th edition, 2013.

III B. Tech. – I Semester

(20BT50505) SOFTWARE PROJECT MANAGEMENT

(Professional Elective - 1)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	3	-	-	3

PRE-REQUISITES: A Course on "Software Engineering"

COURSE DESCRIPTION: Software project management and its importance, Plans, methods and methodologies, Software processes and process models, Stepwise project planning, Software effort estimation, Cost estimation, Activity planning, Plan models, Critical path and critical activities, Risk management, Resource allocation, Monitoring and control, Managing people, Software quality.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Apply knowledge of software project management, project plans, process models for efficient implementation and completion of projects.

CO2. Estimate effort for the project to assign and schedule available resources in the most effective and economical way possible.

CO3. Develop network models for sequences of activities in a project for effective project management.

CO4. Identify the risk factors, monitor the progress and quality of projects to take mitigating actions.

CO5. Recognize the need for organizational behavior, teamwork and communication to improve the performance on projects.

DETAILED SYLLABUS:

UNIT–I: INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT AND PROJECT PLANNING (11 periods)

Introduction to Software Project Management: Importance of software project management, Defining project, Software projects versus other types of project, Contract management and technical project management, Activities covered by software project management, Plans, methods and methodologies, Categorizing software projects, Project charter, Stakeholders, Setting objectives, The business case, Project success and failure, Management and management control, Project management life cycle, Traditional versus modern project management practices.

Project Planning: Step wise project planning, Select project, Identify project scope and objectives, Identify project infrastructure, Analyze project characteristics, Identify project products and activities, Estimate effort for each activity, Identify activity risks, Allocate resources, Review/publicize plan, Execute plan/lower levels of planning.

UNIT–II: PROJECT APPROACH AND EFFORT ESTIMATION (9 periods)

Selection of Project Approach: Build or buy, Choosing methodologies and technologies, Software processes and process models, Choice of process models, Spiral model, Software prototyping, Incremental delivery, Agile methods.

Software Effort Estimation: Effort estimation, Problems with over and under estimates, Basis for software estimating, Software effort estimation techniques, Bottomup estimating, The top-down approach and parametric models, Albrecht function point analysis, COCOMO II.

UNIT–III: ACTIVITY PLANNING AND RISK MANAGEMENT (8 periods)

Activity Planning: Objectives of activity planning, Projects and activities, Sequencing and scheduling activities, Network planning models, Formulating a network model, Adding the time dimension, The forward pass, The backward pass, Identifying the critical path and critical activities, Activity float.

Risk Management: Risk, Categories of risk, Risk management approaches, A framework for dealing with risk, Risk identification, Risk assessment, Risk planning, Risk management, PERT technique, Monte Carlo simulation.

UNIT–IV: RESOURCE ALLOCATION, MONITORING AND CONTROL (9 periods)

Resource Allocation: Nature of resources, Identifying resource requirements, Scheduling resources, Creating critical paths, Publishing resource schedule, Cost schedules, Scheduling sequence.

Monitoring and Control: Creating framework, Review, Visualizing progress, Cost monitoring, Earned value analysis, Prioritizing monitoring, Getting the project back to target, Change control, Software configuration management.

UNIT–V: MANAGING PEOPLE, QUALITY IN SOFTWARE ENVIRONMENT

(8 periods)

Managing People: Organizational behavior, Oldham-Hackman job characteristics model, Ethical and professional concerns, Working in teams – Decision making, Organization and team structures, Dispersed and virtual teams, Communication genres and plans, Leadership.

Software Quality: The place of software quality in project planning, Importance of software quality, Defining software quality, Software quality models, ISO 9126, Process capability models, Quality plans.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOK:

1. Bob Hughes, Mike Cotterell, Rajib Mall, *Software Project Management*, 6th Edition, McGraw Hill, 2018.

REFERENCE BOOKS:

1. Michele Sliger and Stacia Broderick, *The Software Project Manager's Bridge to Agility*, Addison-Wesley, 2008.
2. S.A. Kelkar, *Software Project Management: A Concise Study*, PHI, 2012.
3. Pankaj Jalote, *Software Project Management in Practice*, Pearson, 2002.

ADDITIONAL LEARNING RESOURCES:

- Richard H.Thayer, Software Engineering Project Management, IEEE Computer Society, 2004.
- Information Technology and Project Management, Schwalbe, Thomson Learning.
- https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs70/

III B. Tech. – I Semester

(20BT41531) DESIGN AND ANALYSIS OF ALGORITHMS LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	-	3	1.5

PRE-REQUISITES: A Course on "Design and Analysis of Algorithms"

COURSE DESCRIPTION: Divide and conquer; Quick Sort; Merge Sort; Kruskal's Algorithm; Prim's Algorithm; Dijkstra's Algorithm; Dynamic Programming; Greedy method; Back tracking method; Floyd's Algorithm.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Analyze the performance of Merge sort and quick sort algorithms using divide and conquer technique.

CO2. Develop algorithms to solve knapsack problem using greedy and dynamic programming methods.

CO3. Devise solutions for finding minimum cost spanning tree by using kruskal's and prim's algorithms.

CO4. Solve different shortest path problems by applying Floyd's and Dijkstra's algorithms.

CO5. Implement algorithms to solve real world problems using Dynamic Programming and backtracking methods.

CO6. Work independently and communicate effectively in oral and written forms.

LIST OF EXERCISES:

1. Sort a given set of n integer elements using Quick Sort and merge sort methods and compute its time complexities. Run the programs for varied values of n >1000 and record the time taken to sort. Plot a graph of the time taken versus n for both algorithms. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide and conquer method works along with its time complexity analysis: worst case, average case and best case.

2. Write a program to implement knapsack problem using greedy method.

3. a) Write a program to find minimum cost spanning tree using Kruskal's Algorithm.

   b) Write a program to find minimum cost spanning tree using Prim's Algorithm.

4. Write a program to find shortest paths to other vertices using Dijkstra's algorithm from a given vertex in a weighted connected graph.

5. Write a program to implement 0/1 Knapsack problem using Dynamic Programming method.

6. Write a program to implement All-Pairs Shortest Paths problem using Floyd's algorithm.

7. Write a program to implement Travelling Sales Person problem using Dynamic programming method.

8. Write a program to implement backtracking algorithm for the N-queens problem.

9. Write a program to find a subset of a given set S = {Sl, S2,.....,Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.

10. Write a program to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.

REFERENCE BOOKS:

1. Ellis Horowitz, Satraj Sahni and Rajasekharam, *Fundamentals of Computer Algorithms*, Galgotia Publications Pvt. Ltd, New Delhi, 2nd Edition, 2007.

SOFTWARE/TOOLS USED:

Software: JDK 1.8

Operating System: Windows/ Linux

ADDITIONAL LEARNING RESOURCES:

- NPTEL course on Design and Analysis of Algorithms
  URL: https://nptel.ac.in/courses/106/101/106101060/

III B. Tech. – I Semester

(20BT50532) PYTHON FOR DATA SCIENCE LAB

(Skill Oriented Course)

(Common to CSE and IT)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	1	2	2

PRE-REQUISITES: A Course on "Python Programming Lab"

COURSE DESCRIPTION: Hands on practice on the concepts Python for data science – Computations using NumPy, Data manipulation using Pandas dataframes, Interacting with Web APIs and databases, Data cleaning and preparation, Data wrangling, Plotting and visualization, Handling time series data, Web scraping.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Demonstrate efficient storage and data handling methods in NumPy to perform mathematical computations vital for data science.

CO2. Apply data preparation and exploration methods using Pandas to perform data manipulation.

CO3. Create data visualization using charts, plots and histograms to identify trends, patterns and outliers in data importing Matplotlib and Seaborn.

CO4. Develop methods to analyze and interpret time series data to extract meaningful statistics.

CO5. Work independently to solve problems with effective communication.

LIST OF EXERCISES:

Arrays and vectorized computation using NumPy - The NumPy ndarray, Fast element-wise array functions, Linear algebra, Pseudorandom number generation, Process of exploring data, Pandas data structures – Series, Data frame, Index objects; Essential functionality, Summarizing and computing descriptive statistics - Correlation and covariance, Unique values, Value counts and membership; Data loading, Storage and file formats - Reading and writing data in text format, Binary data formats, Interacting with web APIs, Interacting with databases.

1. Array Computations using NumPy
   a. Perform arithmetic operations using array.
   b. Perform slicing and indexing on multi-dimensional arrays.
   c. Compute arithmetic mean, standard deviation, variance, percentile, minimum and maximum, cumulative sum and product using statistical functions in NumPy.

2. Linear Algebra and Random Number generation using linalg and random module in NumPy
   a) Compute dot product, vector product and inner product of two arrays.
   b) Perform matrix operations such as multiplication, determinant, sum of diagonal elements and inverse.

c) Compute eigenvalues, eigenvectors and singular value decomposition for a square matrix.

d) Generate random samples from uniform, normal, binomial, chi-square and Gaussian distributions using numpy.random functions.

3. Data Manipulation using pandas

a) Create DataFrame from List, Dict, List of Dicts and perform operations such as column selection, addition, deletion and row selection.

b) Create a DataFrame and perform descriptive statistics functions such as sum, mean, median, mode, standard deviation, skewness, kurtosis, cumulative sum, cumulative product and percent changes.

c) Implement the computation of correlation and covariance by considering the DataFrames of stock prices and volumes obtained from Yahoo Finance! Using pandas-datareader package.

4. Working with different data formats using pandas

a) Perform reading and writing data in text format using read_csv and read_table considering any online dataset in delimited format (CSV).

b) Perform reading, writing and parsing data in JSON (Javascript Object Notation) format using read_json.

c) Perform reading and writing of Microsoft Excel Files (xslx) using read_excel.

5. Interacting with Web APIs and Databases

a) Predict the last 30 GitHub issues for pandas using request and response object's json method. Move the extracted data to DataFrame and extract fields of interest. (Use url: https://api.github.com/repos/pandas-dev/pandas/issues)

b) Connect to any relational database using corresponding SQL drivers and perform operations such as table creation, populating the table, selecting data from table, moving data from table to DataFrame, updating records and deleting records in a table.

Handling missing data, Data transformation, Vectorized string functions in pandas; Data wrangling: Join, combine and reshape - Hierarchical indexing, Combining and merging datasets, Reshaping and pivoting.

6. Data Cleaning and Preparation

a) Perform data cleaning by creating a DataFrame and identifying missing data using NA(Not Available) handling methods, filter out missing data using dropna function, fill the missing data using fillna function and remove duplicates using duplicated and drop_duplicates functions.

b) Perform data transformation by modifying set of values using map and replace method and create transformed version of original dataset without modification using rename method.

c) Create a DataFrame with normally distributed data using random sampling and detect possible outliers.

7. Data Wrangling

   a) Perform hierarchical indexing by creating a series with a list of lists (or arrays) as the index, select subsets of data at outer and inner levels using partial indexing.

   b) Rearrange the tabular data with hierarchical indexing using unstack and stack method.

Plotting and visualization - A brief Matplotlib API Primer, Plotting with pandas and seaborn, Other python visualization tools; Date and time data types and tools, Time series basics, Date ranges, Frequencies, and shifting, Time zone handling, Periods and period arithmetic, Resampling and frequency conversion – Downsampling, upsampling and interpolation; Web scraping using beautiful soup.

8. Perform Data Visualization with Matplotlib and SeaBorn considering online dataset for processing.

   a) Create a Line Plot by setting the title, axis labels, ticks, ticklabels, annotations on subplots and save to a file.

   b) Create Bar Plots using Series and DataFrame index.

      i) Create bar plots with a DataFrame to group the values in each row together in a group in bars side by side for each value.

      ii) Create stacked bar plots from a DataFrame.

   c) Create Histogram to display the value frequency and Density Plot to generate continuous probability distribution function for observed data.

   d) Create Scatter Plot and examine the relationship between two one-dimensional data series.

   e) Create Box plots to visualize data with many categorical variables.

9. Time Series Analysis

   a) Create time series using datetime object in pandas indexed by timestamps.

   b) Use pandas.date_range to generate a DatetimeIndexwith an indicated length.

   c) Generate data ranges by setting time zone, localize time zone and convert to particular time zone using tz_convert and combine two different time zones.

   d) Perform period arithmetic such as adding and subtracting integers from periods and construct range of periods using period_range function.

10. Web Scraping using Beautiful Soup

    Extract product reviews from Amazon website and save to a file. Perform Exploratory Data Analysis on extracted product reviews.

      i) Generate WordCloud for all reviews, positive and negative reviews.

      ii) Plot the distribution of stopwords, numerics, wordcount values, charactercount values and average wordlength.

      iii) Display the sentiment value using nltk and vader.

      iv) Create Scatter Intensity Plot of Sentiments.

REFERENCE BOOKS:

1. Wes McKinney, *Python for Data Analysis*, O'Reilly, 2nd Edition,2017.
2. Sinan Ozdemir, *Principles of Data Science*, Packt Publishers, 2nd Edition, 2018.
3. Rachel Schutt, Cathy O'Neil, *Doing Data Science: Straight Talk from the Frontline*, O'Reilly, 2014.

III B. Tech. – I Semester

(20BT5MC01) PROFESSIONAL ETHICS

(Mandatory Course)

(Common to CSE, CSE(AI), CSE(DS), CSBS, CSSE and IT)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	-	30	2	-	-	-

PRE-REQUISITES: --

COURSE DESCRIPTION: Engineering Ethics; Professional Ideals and Virtues; Engineering as Social Experimentation; Responsibilities and Rights; Global Issues.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Demonstrate knowledge of Engineering Ethics, Senses of engineering ethics, Moral dilemmas and theories in professional engineering practice

CO2. Analyze the concepts of Professional ideals to assess and to address societal, health, safety, legal and cultural issues in discharging the professional responsibilities

CO3. Apply the reasoning informed by the various aspects of Code of Ethics and its provisions to assess societal issues and carry out Professional responsibilities effectively

CO4. Practice Collegiality considering conflict of interests to safeguard professional rights in professional engineering practice.

CO5. Provide professional engineering solutions considering distinct ethics to address global issues.

DETAILED SYLLABUS:

UNIT-I: ENGINEERING ETHICS (5 periods)

Scope and aim of engineering ethics, Senses of engineering ethics, Variety of moral issues, Types of inquiry, Moral dilemmas, Moral autonomy-Kohlberg's theory, Gilligan's theory, Consensus and controversy.

UNIT-II: PROFESSIONAL IDEALS AND VIRTUES (6 periods)

Theories about virtues, Professions, Professionalism, Characteristics, Expectations, Professional responsibility, Integrity, Self-respect, Sense of responsibility, Self-interest, Customs and religion, Self-interest and ethical egoism, Customs and ethical relativism, Religion and divine command ethics, Use of ethical theories, Resolving moral dilemmas and moral leadership.

UNIT-III: ENGINEERING AS SOCIAL EXPERIMENTATION (7 periods)

Engineering as experimentation, Similarities to standard experiments, Learning from the past and knowledge gained, Engineers as responsible experimenters, Conscientiousness, Moral autonomy and accountability, The challenger case, Codes of ethics and limitations, Industrial standards, Problems with the law of engineering.

UNIT-IV: RIGHTS AND RESPONSIBILITIES OF AN ENGINEER (6 periods)

Collegiality and loyalty, Respect for authority, Collective bargaining, Confidentiality, Conflict of interests, Occupational crime, Rights of engineers, Professional rights, Whistle-blowing, The BART case, Employee rights and discrimination.

UNIT-V: GLOBAL ISSUE (6 periods)

Multinational corporations, Professional ethics, Environmental ethics, Computer ethics, Engineers as consultants, Witnesses, Advisors and Leaders, Engineers as Managers, Managerial ethics applied to Engineering Profession, moral leadership.

Total Periods: 30

Topics for self-study are included in the lesson plan

TEXT BOOKS:

1. Mike W. Martin and Roland Schinzinger, *Ethics in Engineering*, Tata McGraw-Hill, 3rdedition, 2007.
2. Govindarajan, M., Nata Govindarajan, M., Natarajan, S. and Senthil kumar, V.S., *Engineering Ethics*, Prentice Hall of India, 2004.

REFERENCE BOOKS:

1. S. Kannan and K. Srilakshmi, Human Values and Professional Ethics, Taxmann Allied Services Pvt. Ltd., 2009.
2. Edmund G. Seebauer and Robert L. Barry, Fundamental of Ethics for Scientists and Engineers, Oxford University Press, 2001.

IV B. Tech. – I Semester

(20BT70501) COMPUTER VISION

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	3	-	-	3

PRE-REQUISITES: Courses on "Transformation Techniques and Linear Algebra", "Machine Learning"

COURSE DESCRIPTION: Concepts of low-level vision, Image filtering operations, Masking, Thresholding techniques, Edge Detection, Dilation and erosion, Background subtraction, Shot boundary detection, Interactive segmentation, Clustering based segmentation, Texture, Classification, Overfitting, Receiver Operator curves, Object detection and recognition and Information Retrieval methods.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Analyze image filtering operations to enhance image quality.

CO2. Apply threshold techniques, morphological process and region growing methods for edge detection in images.

CO3. Develop clustering based segmentation solutions for image synthesis.

CO4. Synthesize and evaluate classification procedures for texture and feature analysis.

CO5. Select and apply appropriate techniques for object recognition and detection in computer vision based applications.

DETAILED SYLLABUS:

UNIT–I: INTRODUCTION AND IMAGE ENHANCEMENT (8 periods)

The nature of the vision, Low-level vision – Gray scale versus color, Image processing operations; Basic image filtering operations – Gaussian smoothing, Median filters, Mode Filters, Rank Order Filters, Sharp and Unsharp masking.

UNIT–II: THRESHOLDING AND EDGE DETECTION (10 periods)

Region-growing methods, Thresholding, Adaptive thresholding, Threshold selection – Variance-based thresholding, Entropy-based thresholding, Maximum likelihood thresholding; Global valley approach to thresholding; Edge Detection – Template Matching Approach, 3×3 Template Operators, Canny Operator, Laplacian Operator; Dilation and erosion in binary images – Properties of dilation and erosion operators, Closing and opening.

UNIT–III: SEGMENTATION BY CLUSTERING (9 periods)

Grouping and gestalt, Important applications – Background subtraction, Shot boundary detection, Interactive segmentation, Forming imaging regions; Image segmentation by clustering pixels, Segmentation, clustering and graphs – Terminology and facts for graphs, Agglomerative clustering with a graph, Divisive clustering with a graph, Normalized cuts.

UNIT–IV: CLASSIFICATION AND DETECTION OF OBJECTS (11 Periods)

Texture – Spots and bars, Representation, Synthesizing textures and filling holes in images, Shape from texture; Learning to classify – Using loss to determine decisions, Training error, test error and overfitting, Regularization, Error rate and Cross-validation, Receiver operating curves; Classifying images – Classifying images of single objects; Detecting objects in images – The sliding window method.

UNIT–V: OBJECT RECOGNITION, APPLICATIONS (7 periods)

Object Recognition – Categorization, Selection, Feature questions, Geometric questions, Semantic questions; Applications – Classifying materials, Classifying scenes, Tracking people.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOKS:

1. David A. Forsyth, Jean Ponce, *Computer Vision: A Modern Approach*, Pearson, 2nd Edition, 2012.
2. E. R. Davies, *Computer and Machine Vision: Theory, Algorithms, Practicalities*, Elsevier, 5th Edition, 2017.

REFERENCE BOOKS:

1. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Pearson, 4th Edition, 2018.
2. William K. Pratt, *Digital Image Processing*, Wiley, 4th Edition, 2006.

ADDITIONAL LEARNING RESOURCES:

1. https://slideplayer.com/slide/5158896/
2. www.scs.carleton.ca/~c_shu/Courses/comp4900d/notes/PPT/lect1_intro.ppt

IV B. Tech. – I Semester

(20BT71206) GAME DEVELOPMENT

(Professional Elective – 5)

(Common to CSE, CSE(AI), CSE(DS) and IT)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	3	-	-	3

PRE-REQUISITES: A Course on "Object Oriented Programming through Java"

COURSE DESCRIPTION: Third Dimension, Prototyping and Scripting Basics; Creating the Environment, Player Characters and Further Scripting; Interactions, Collection, Inventory, and HUD (heads up display); Instantiation and Rigidbodies, Particle Systems; Designing Menus, Animation Basics, and Building.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Demonstrate knowledge on Third Dimension, Prototyping and Scripting.

CO2. Build game environment, player characters and script for game development.

CO3. Analyze the interactions between players, understand the Collections and Inventory for game development.

CO4. Develop games using rigidbodies and particle systems.

CO5. Design menus, animations and publish the games.

DETAILED SYLLABUS:

UNIT-I:THIRD DIMENSION, PROTOTYPING AND SCRIPTING BASICS (9 Periods)

Third Dimension: Getting to grips with 3D, Rigidbody physics, Essential Unity concepts, The interface.

Prototyping and Scripting Basics: first Unity project, A basic prototyping environment, Introducing scripting, Understanding Translate, Testing the game so far, Storing with prefabs, Using Instantiate() to spawn objects.

UNIT-II: CREATING THE ENVIRONMENT, PLAYER CHARACTERS AND FURTHER SCRIPTING (9 Periods)

Creating the Environment: Designing the game, Using the terrain editor, The terrain toolset, Creating the island—sun, sea, and sand.

Player Characters and Further Scripting: Working with the Inspector, Anatomy of a character, Deconstructing the First Person Controller object, Further scripting, Full example, Inter-script communication and Dot Syntax, Scripting for character movement.

UNIT-III: INTERACTIONS, COLLECTION, INVENTORY, AND HUD (HEADS UP DISPLAY) (9 Periods)

Interactions: External modeling applications, Setting up the outpost model, Adding the outpost, Collisions and triggers, Ray casting, Opening the outpost.

Collection, Inventory, and HUD: Creating the power cell prefab, Scattering power cells, Restricting outpost access, Displaying the power cell HUD.

UNIT-IV: INSTANTIATION AND RIGIDBODIES, PARTICLE SYSTEMS (9 Periods)

Instantiation and Rigidbodies: Utilizing instantiation, Rigidbodies, Making the mini-game.

Particle Systems: particle system, Creating the task, Testing and confirming.

UNIT-V: DESIGNING MENUS, ANIMATION BASICS, AND BUILDING (9 Periods)

Designing Menus: Interfaces and menus, Creating the menu with GUI Textures and mouse events, Creating the menu with the Unity GUI class and GUI skins

Animation Basics: Game win sequence.

Building: Build options, Build Settings, Player Settings, Quality Settings, Building the game.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOKS:

1. Will Goldstone, *Unity 3.x Game Development Essentials Game development with C# and Javascript*, PACKT Publishing, Second Edition, 2011.

REFERENCE BOOKS:

1. Sue Blackman, *Beginning 3D Game Development with Unity 4*, Apress Publisher, Second Edition, 2013.
2. Mike Geig, *Unity Game Development in 24 Hours*, Pearson Education, First Edition, 2013.

ADDITIONAL LEARNING RESOURCES:

1. Making your first game - https://learn.unity.com/tutorial/your-first-game#
2. Game Development Tutorial using Unity 3D - https://www.studytonight.com/game-development-in-2D/

IV B. Tech. – I Semester

(20BT70506) SERVICE ORIENTED ARCHITECTURE AND MICROSERVICES ARCHITECTURE

(Professional Elective – 5)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	3	-	-	3

PRE-REQUISITES: A Course on "Software Engineering"

COURSE DESCRIPTION: Software engineering principles, Agile with DevOps, Software architecture, Architecting process, SOA and MSA basics, Enterprise-wide SOA, Service oriented analysis and design, Software platforms, SOA applications, Technologies for SOA; SOA governance and implementation, SOA best practices, Trends in SOA and MSA, Big data and SOA, Cloud and MSA, Mobile and MSA.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Identify suitable software architecture by analyzing software requirements, engineering principles and development methodologies.

CO2. Apply the architecting process for developing SOA and MSA for software applications.

CO3. Design service-oriented architecture on enterprise level perspective by understanding the architectural aspects and types of services.

CO4. Use software platforms, technologies, best practices, strategy development and governance for providing SOA solutions.

CO5. Demonstrate knowledge on SOA and MSA applications and hybrid service models to address the business needs of an organization.

DETAILED SYLLABUS:

UNIT–I: SOFTWARE ARCHITECTURE (9 periods)

Software Engineering Principles: Software development life cycle, Project management practice, Software engineering project, Critical success factors for software projects.

Agile with DevOps: Agile development methodologies, Emergence of DevOps, DevOps application delivery, Continuous integration, Continuous delivery, Continuous deployment, DevOps – putting it all together.

Software Architecture: Need, Objectives, Types of Information Technology architecture, Architectural patterns and styles.

Architecting Process: Architectural considerations, Architecting process for software applications, High level architecture, Solution architecture, Detailed design.

UNIT–II: SERVICE ORIENTED ARCHITECTURE AND MICROSERVICES ARCHITECTURE (9 periods)

SOA and MSA Basics: Service orientation, Evolution, Basic concepts, Drivers and dimensions of SOA, Conceptual model of SOA, Standards and guidelines for SOA, Emergence of MSA.

Enterprise-Wide SOA: Considerations, Strawman architecture, Reference architecture, Object oriented analysis and design process, Service oriented analysis and design process, SOA methodology for enterprise.

Service Oriented Analysis and Design: Need for models, Principles of service design, Non-functional properties for services, Design of activity services, Design of data services, Design of client services, Design of business process services.

UNIT–III: PLATFORMS, APPLICATIONS, TECHNOLOGIES FOR SERVICE ORIENTED ARCHITECTURE (11 periods)

Software Platforms: Software applications in an enterprise, Packaged applications, Custom software applications, Software platforms, Java platform enterprise edition, .NET Microsoft application platform.

Service-Oriented Applications: Considerations, Patterns, Pattern based architecture for service-oriented applications, Composite applications, Composite application programming model.

Technologies for SOA: Technologies for service enablement, Technologies for service integration, Technologies for service orchestration.

UNIT–IV: SERVICE ORIENTED ARCHITECTURE GOVERNANCE AND PRACTICES (7 periods)

SOA Governance and Implementation: Strategic architecture governance, Service design-time governance, Service run-time governance, Approach for enterprise-wide SOA implementation.

SOA Best Practices: SOA Strategy best practices, SOA Development best practices, SOA Governance best practices.

UNIT–V: TRENDS IN SERVICE ORIENTED ARCHITECTURE AND MICROSERVICES ARCHITECTURE (9 periods)

Trends in SOA and MSA: Services model for cloud and mobile solutions, API adoption on the rise, Challenges and takeaways from SOA implementations, Architecture trend, Microservices architecture in action.

Big Data and SOA: Data concepts, Big data and its characteristics, Technologies for big data, Service-orientation for big data solutions.

Cloud and MSA: Cloud services, Hybrid cloud services, Considerations for hybrid cloud services, Cloud services and MSA, MSA for SMAC solutions.

Mobile and MSA: Mobile technologies, Types of mobile applications, MSA for mobile solutions.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOK:

1. Shankar Kambhampaty, *Service-Oriented Architecture & Microservices Architecture, For Enterprise, Cloud, Big Data and Mobile*, 3rd Edition, Wiley, 2018.

REFERENCE BOOKS:

1. Thomas Erl, *Service-Oriented Architecture: Analysis and Design for Services and Microservices*, 2nd Edition, Prentice Hall, 2016.
2. Irakli Nadareishvili, Ronnie Mitra, Matt McLarty, Mike Amundsen, *Microservice Architecture: Aligning Principles, Practices, and Culture*, O'Reilly, 2016.

IV B. Tech. – I Semester

(20BT70507) NATURE INSPIRED ALGORITHMS

(Interdisciplinary Elective – 2)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	3	-	-	3

PRE-REQUISITES: Courses on "Differential Equations and Multivariable Calculus", "Optimization Techniques"

COURSE DESCRIPTION: Optimization, Classical optimization techniques, Nature inspired algorithms, Genetic algorithm, Particle swarm optimization, Ant colony optimization, Bee colony optimization, Cuckoo search algorithm, Firefly algorithm, Bat algorithm, Gray wolf optimization, Elephant herding optimization, Applications of nature inspired algorithms.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Demonstrate knowledge on optimization and classical optimization techniques to find optimal solutions for a given problem.

CO2. Analyze the key components and mathematical aspects of nature inspired algorithms.

CO3. Design efficient solutions for optimization problems using nature inspired algorithms.

CO4. Investigate the applications of nature inspired algorithms to solve wide range of optimization problems.

DETAILED SYLLABUS:

UNIT–I: INTRODUCTION TO OPTIMIZATION (9 periods)

Introduction to Optimization: Fundamentals of optimization, Types of optimization problems, Examples of optimization, Formulation of optimization problems, Classification of optimization algorithms, Traveling salesman problem, Knapsack problem.

Classical Optimization Techniques: Mathematical model of optimization, Linear programming – Simplex method, Revised simplex method, Kamarkar's method, Duality theorem, Decomposition principle, Transportation problem; Nonlinear Programming – Quadratic programming, Geometric programming; Dynamic programming, Integer programming, Stochastic programming, Lagrange multiplier method.

UNIT–II: NATURE INSPIRED ALGORITHMS AND GENETIC ALGORITHM (8 periods)

Nature Inspired Algorithms: Traditional vs nature inspired algorithms, Bioinspired algorithms, Swarm intelligence, Metaheuristics, Diversification and intensification, No free lunch theorem, Parameter tuning and control, Algorithm.

Genetic Algorithm: Basics, Genetic operators, Example of GA, Algorithm, Schema theory, Prisoner's dilemma problem, Variants and hybrids of GA.

UNIT–III: PARTICLE SWARM, ANT COLONY, BEE COLONY AND CUCKOO SEARCH OPTIMIZATION ALGORITHMS (10 periods)

Particle Swarm Optimization: Swarm behavior, Algorithm, Variants of algorithm.

Ant Colony Optimization: Ant colony characteristics, Ant colony optimization – Travelling salesman problem, algorithm; Variants of algorithm.

Bee Colony Optimization: Honey bee characteristics, Algorithm, Variants of algorithm.

Cuckoo Search Algorithm: Cuckoo bird behavior, Levy flights, Algorithm, Variants of algorithm.

UNIT–IV: FIREFLY, BAT, GRAY WOLF AND ELEPHANT HERDING OPTIMIZATION ALGORITHMS (9 periods)

Firefly Algorithm: Firefly behavior and characteristics, Algorithm, Variants and applications.

Bat Algorithm: Behavior of bats in nature, Algorithm, Variants and applications.

Gray Wolf Optimization: Gray wolf characteristics, Gray wolf optimization, Variants and applications.

Elephant Herding Optimization: Elephant herding behavior, Algorithm, Pseudocode, Variants of the algorithm.

UNIT–V: APPLICATIONS OF NATURE INSPIRED ALGORITHMS (9 periods)

Image processing, Classification, clustering and feature selection, Traveling salesman problem, Vehicle routing, Scheduling, Software testing, Deep belief networks, Swarm robots, Data mining and deep learning – Clustering, Support vector machines, Artificial neural networks, Optimizers for machine learning, Deep learning.

Total Periods: 45

Topics for self-study are provided in the lesson plan

TEXT BOOKS:

1. A. Vasuki, *Nature-Inspired Optimization Algorithms,* CRC Press, 2020.
2. Xin-She Yang, *Nature-Inspired Optimization Algorithms,* Elsevier, 2nd Edition, 2020.

REFERENCE BOOKS:

1. Xin-She Yang, Xing-Shi He, *Mathematical Foundations of Nature-Inspired Algorithms*, Springer, 2019.
2. George Lindfield, John Penny, *Introduction to Nature-Inspired Optimization,* Elsevier, 2017.

ADDITIONAL LEARNING RESOURCES:

- Xin-She Yang, *Nature-Inspired Computation and Swarm Intelligence: Algorithms, Theory and Applications,* Elsevier, 2020.
- Hema Banati, Shikha Mehta, Parmeet Kaur, *Nature-Inspired Algorithms for Big Data Frameworks*, IGI Global, 2019.

IV B. Tech. – I Semester

(20BT70531) COMPUTER VISION LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	-	3	1.5

PRE-REQUISITES: Courses on "Python Programming", "Computer Vision"

COURSE DESCRIPTION: Image Filtering Operations, Thresholding Techniques, Edge Detection, Dilation and Erosion, Background Subtraction, Clustering based segmentation, Texture, Classification, Receiver Operator Curves, Object Recognition and Information Retrieval methods.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Analyze and implement computer vision techniques for image enhancement, synthesis and segmentation.

CO2. Synthesize and implement classification procedures for texture and feature analysis.

CO3. Develop applications like face recognition system, image classification system using computer vision methods.

CO4. Work independently to solve problems with effective communication.

LIST OF EXERCISES:

1. Implement the following image pre-processing commands:
    a. To read an image
    b. To show an image
    c. Convert RGB to Gray Scale
    d. Read RGB values of a pixel
    e. Convert Gray Scale to Binary
    f. Perform Image Crop
    g. Perform Image Resize
    h. Perform Image Rotation
    i. Histogram Equalization

2. Implement the following image filtering techniques:
    a. Median Filters
    b. Mode Filters

3. Implement Image sharpening using 2-D Laplacian high pass filter in spatial domain.

4. Implement Edge detection in a segmented binary image using Canny edge detector.

5. Implement Morphological image operations – erosion, dilation, opening & closing to correct the damaged Digital photo image.

6. Implement Color image processing to differentiate objects in a digital image.

(Image Database Prints and Photographs online Catalog

url: http://www.loc.gov/pictures)

7. Implement Image segmentation using K-means clustering method in a Fruit basket application.

   (Image Database: CVOnline: Image Databases

   url: http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm)

8. Implement Image segmentation using Watershed transform in any Scenery photo.

   (Image Database: CVOnline: Image Databases

   url: http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm)

9. Implement a program for Features Identification in Fingerprint Recognition System.

   (Image Database: FVC2006 fingerprint database

   url: http://atvs.ii.uam.es/atvs/fvc2006.html)

10. Implement a program using Texture based Image Classification, find the accuracy level using Receiver Operating curve.

    (Image Database: Biometric Spatial Databases and Software –

    url: https://www.nist.gov/itl/iad/image-group/resources/biometric-special-databases-and-software)

11. Implement a Sliding window method and detect the objects present in the Video Surveillance system.

    (Image Database: VIRAT Video Dataset, url: https://viratdata.org/)

REFERENCE BOOKS:

1. Will Ballard, *Hands-on Deep Learning for Images with Tensorflow: Build Intelligent Computer Vision Applications using Tensorflow and Keras*, Packt Publishing, 2018.
2. Ahmed Fawzy Mohamed Gad, *Practical Computer Vision Applications using Deep Learning with CNNs: With Detailed Examples in Python using Tensorflow and Kivy*, Apress, 2018.
3. Abhinav Dadhich, *Practial Computer Vision: Extract Insightful Information from Images using Tensorflow, Keras, OpenCV*, Packt, 2018.

SOFTWARE/TOOLS:

- OpenCV
- Tensorflow
- Virtual Labs (Image Processing Lab - http://cse19-iiith.vlabs.ac.in/)

ADDITIONAL LEARNING RESOURCES:

- http://coeosmanabad.com/etc/manual/BE%20DIP%20Lab%20Manual.pdf
- https://jnec.org/Lab-manuals/CSE/CSE1/TE-Part-1/DIP.pdf
- http://gn.dronacharya.info/CSEDept/Downloads/Labmanuals/DIP-Lab-Manual.pdf
- http://titagartala.ac.in/wp-content/uploads/2018/12/Digital-Lab-Processing-Lab-Manual.pdf

IV B. Tech. – I Semester

(20BT70532) R PROGRAMMING LAB

(Skill Oriented Course)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	1	2	2

PRE-REQUISITES: --

COURSE DESCRIPTION: Hands on experience on Introduction to R, Data structures, Control statements, Functions, Object oriented programming, Statistical analysis, Importing data into R, Data visualization.

COURSE OUTCOMES: *After successful completion of the course, students will be able to:*

CO1. Apply R programming constructs to store and manipulate datasets.

CO2. Develop modules using R programming constructs to solve statistical problems.

CO3. Perceive data models to perform descriptive and inferential statistical analysis to identify trends, patterns in data.

CO4. Create effective visualization using Histograms, Barplots, Boxplots, Scatterplots for exploratory data analysis.

CO5. Work independently to solve problems with effective communication.

LIST OF EXERCISES:

Datatypes, Variables, Operators, Data structures – Vectors, Arrays, Matrices, Lists, Data frames; Object oriented programming – S3, S4 classes; Selection statements – if statement, if else statement, switch statement; Iterative statements – For loop, While loop, Repeat loop, Nested loops; Functions – Creating functions, Default values for arguments, Return values, Environment and scope issues, Recursion.

1. Create the vectors:

   (a) (1, 2, 3, . . . , 19, 20)

   (b) (20, 19, . . . , 2, 1)

   (c) (1, 2, 3, . . . , 19, 20, 19, 18, . . . , 2, 1)

   (d) (4, 6, 3) and assign it to the name tmp.

   For parts (e), (f) and (g) look at the help for the function rep.

   (e) (4, 6, 3, 4, 6, 3, . . . , 4, 6, 3) where there are 10 occurrences of 4.

   (f) (4, 6, 3, 4, 6, 3, . . . , 4, 6, 3, 4) where there are 11 occurrences of 4, 10 occurrences of 6 and 10 occurrences of 3.

   (g) (4, 4, . . . , 4, 6, 6, . . . , 6, 3, 3, . . . , 3) where there are 10 occurrences of 4, 20 occurrences of 6 and 30 occurrences of 3.

2. a) Write R code that will generate a vector with the following elements.

   "aa" "ba" "ca" "da" "ea" "ab" "bb" "cb" "db" "eb" "ac" "bc" "cc" "dc" "ec" "ad" "bd" "cd" "dd" "ed" "ae" "be" "ce" "de" "ee"

b) Write a R program to create a Dataframes which contain details of 5 employees and display summary of the data.

3. a) Create a vector of a data set and treat it as an object. Using the vector and object perform (.) dot product and (x) cross product. Take your own data.

b) "Fizzbuzz" is a simple programming challenge often used at interviews to test very basic programming skill. Your goal is the following: for the numbers 1 to 100, print "fizz" if the number is a multiple of 3, "buzz" if the number is a multiple of 5, "fizzbuzz" if the number is a multiple of both 3 and 5, and simply print the number otherwise.

4. a) Imagine a high school with 1000 lockers all in a row, numbered 1 to 1000 in order. At the start, all of them are closed. 1000 students are sent, one after the other, to change the state of a set of lockers (from open to closed or closed to open). The first student changes the state of all lockers. The second changes the state of every other one (2, 4, 6, 8, . . .). The third changes the state of every third one (3, 6, 9, 12, . . .). This process continues until all 1000 students have gone. Write a R program to determine which lockers are open at the end of this process?

b) Write a function chomp() that, given a string, removes from the string any occurrence of the character &, as well as the character to the left of each & character. So, for example, your function should return:

```
 > chomp ( " a&c " )
" c "
> chomp ( " a&" )
 " "
> chomp ( " abc " )
" abc "
```

5. a) Write a function which takes a single argument which is a matrix. The function should return a matrix which is the same as the function argument but every odd number is doubled.

b) Write a function that takes an array of numbers x and returns the smallest number in the array.

Importance and applications of statistical learning, Types of data, Types of variables, Frequency distributions, Measures of center – Mean, Median, Mode; Measures of spread – Range, Percentile, Quartiles & Interquartile range, Standard deviation, Variance; Correlation and Covariance.

6. a) Compute descriptive statistics for the data given below.

X: 14, 20, 22, 19, 15, 18, 30, 27

Y: 16, 25, 27, 20, 16, 18, 27, 23

b) Write a R script which will compute the mean and variance of the vector x <- 1:100. Compare with R's internal mean() and var() functions.

7. a) Write a function to compute running medians. Running medians are a simple smoothing method usually applied to time-series. For example, for the numbers 7,5, 2, 8, 5, 5, 9, 4, 7, 8, the running medians of length 3 are 5, 5, 5, 5, 5, 5, 7, 7. The first running median is the median of the three numbers 7, 5, and 2; the second running median is the median of 5, 2, and 8; and so on. Your function should take two arguments: the data (say, x), and the number of observations for each median (say, length).

b) Write a R script to calculate Central tendency and spread by choosing the appropriate dataset.

Importing data into R – Importing data from a CSV file, Importing data from Excel; Data Visualization – Histograms, Barplots, Boxplots, Scatterplots, correlation matrix.

8. a) Write a R program to perform data import/export (.csv, .xlxs) operations using data frames in R.

   b) Write a R program to draw Box plot by choosing appropriate dataset.

9. a) Write a R program to create bell curve of a random normal distribution.

   b) Write a R program to draw Histograms by choosing appropriate dataset.

10. a) Write a R program to draw Scatter plots by choosing appropriate dataset.

    b) Write a R program to design correlation matrix by choosing appropriate dataset.

REFERENCE BOOKS:

1. Sandip Rakshit, *R for Beginners*, McGraw Hill, 2017.
2. Seema Acharya, *Data analytics using R*, McGraw Hill, 2018.