SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)
Sree Sainath Nagar, Tirupati
Department of Information Technology

Supporting Document for 1.1.2

Syllabus Revision carried out in 2020

Program: B.Tech- Information Technology

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)*

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.	20BT11201	Programming For Problem Solving	100	3
2.	20BT31231	Software Engineering Lab	20	6
3.	20BT31232	Programming with Python Lab	100	14
4.	20BT50531	Python for Data Science Lab	100	17
5.	20BT61204	Web Development using Angular and Bootstrap	100	21
6.	20BT61232	AI Tools Lab	100	23
7.	20BT61233	Advanced Web Development Lab	100	25
8.	20BT71209	Digital and Social Media Marketing	100	27
9.	20BT71233	Augmented Reality and Virtual Reality Applications Lab	100	29
10.	20BT712AC	Deep Learning Technologies	100	31
		Average % (A)	92	
		Total No. of courses in the Program (T)	124	
		No. of Courses where syllabus (more than 20% content) has been changed (N)	10	-
		Percentage of syllabus content change in the courses (C) = (A X N) / 100	9.2	
		Percentage of syllabus content change in the program (P) = (C / T) *100	7.4	

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

I B. Tech. – I Semester

(20BT11201) PROGRAMMING FOR PROBLEM SOLVING

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

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

PRE-REQUISITES: ACourse on Basic Mathematics.

COURSE DESCRIPTION: Introduction to C Programming; Operators and Expressions; Input and Output Functions; Control Structures; Problem Solving Aspects; Arrays and Strings; Functions; Pointers; Structures and Unions; File Handling.

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

CO1. Demonstrate knowledge on C programming constructs to develop programs.

CO2.Design algorithms usingproblem-solving techniques for given problems.

CO3. Apply functions and Arrays to enhance reusability and data manipulation.

CO4. Use pointers to manage the memory effectively.

CO5. Apply Structures, Unions and File handling concepts to develop societal applications.

DETAILED SYLLABUS:

UNIT I - INTRODUCTION TO C PROGRAMMING   (09 periods)

Basics of C Programming: Introduction, Structure of a C program, Concept of a variable, Data types in C, Program statement, Declaration, Storing the data in memory, Tokens, Operators and expressions, Lvalues and Rvalues, Type conversion in C.

Input and Output: Basic screen and keyboard I/O in C, Non-formatted input and output, formatted input and output functions.

UNIT II - CONTROL STATEMENTS AND INTRODUCTION TO PROBLEM SOLVING

**   (08 periods)**

Control Statements: Specifying test condition for selection and iteration, Writing test expression, Conditional execution and selection, Iteration and repetitive execution, goto statement, Special control statements, Nested loops.

Introduction to Problem Solving:Algorithms, Flowcharts, Problem solving aspect, Top-down design, Implementation of algorithms, program verification and efficiency of algorithms.

UNIT III - ARRAYS AND STRINGS, FUNCTIONS   (10periods)

Arrays and Strings: One-dimensional array – Declaration, Initialization, Accessing elements, operations; Multi-dimensional arrays – Declaration, Initialization, Working with 2D arrays; Strings – Declaration, Initialization, Printing strings, String input, Character manipulation, String manipulation; Arrays of strings – Initialization, manipulating string arrays.

Functions: Concept of function, Using functions, Call by value mechanism, working with functions, passing arrays to functions, Scope and extent, Storage classes, Recursion.

UNIT IV – POINTERS **(08 periods)**

Pointers in C: Understanding memory addresses, Address operator (&), Pointer – declaration, Initialization, Indirection operator and dereferencing,Void and Null pointers, Use of pointers,Arrays and pointers, Pointers and strings, Pointer arithmetic, Pointers to pointers, Array of pointers, Pointers to an array, Two-dimensional arrays and pointers, Pointers to functions, Dynamic memory allocation.

UNIT V - USER-DEFINED DATA TYPES AND FILES **(10 periods)**

User-Defined Data Types: Structures - Declaration, Accessing the members, Initialization,typedef and its use, Arrays of structures, Arrays within structure, Structures and pointers, Structures and functions; Unions, Enumeration types, Bitfields.

Files in C: Using files in C, Working with text and binary files, Direct File Input and Output, Files of records, Random access to files of records.

Total Periods: 45

Topics for Self-study are provided in the Lesson Plan

TEXT BOOKS:

1. PradipDey and Manas Ghosh, *Programming in C*, Second Edition, Oxford University Press, New Delhi, 2013.
2. R. G. Dromey, *How to Solve it by Computer*, First Edition, Pearson Education, 2013.

REFERENCE BOOKS:

1. Byron S Gottfried and Jitender Kumar Chhabra, *Programming with C*, Fourth Edition, McGraw Hill Education, 2019.
2. YashavantKanetkar, *Let Us C*, Fifteenth Edition, BPB Publications, 2017.
3. E. Balagurusamy, *Programming in C*, Seventh Edition, McGraw Hill Education Pvt, Ltd, New Delhi, 2017.
4. Behrouz A. Forouzan and Richard F. Gilberg, *Computer Science: A Structured Programming Approach Using C*, Third Edition, Cengage Learning, 2008.

ADDITIONAL LEARNING RESOURCES:

- Learn C Programming - https://www.programiz.com/c-programming
- Learn C Programming - https://www.tutorialspoint.com/cprogramming/index.htm

- C Programming Exercises, Practice, Solution - https://www.w3resource.com/c-programming-exercises/

- Basic programming exercises and solutions in C - https://codeforwin.org/2015/05/basic-programming-practice-problems.html

II B.Tech. – I Semester

(20BT31231) SOFTWARE ENGINEERING LAB

(Common to CSSE, IT & CSBS)

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 OBJECTIVES:

- To impart knowledge on the concepts of Unified Modeling Language (UML).
- To develop skills related to identify requirements , SRS preparation and designing software using UML
- To inculcate attitude to solve societal problems using UML models.

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
   1. 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, 2nd 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.

II B. Tech. – I Semester
(19BT31232) **SOFTWARE ENGINEERING LAB**

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:

CO6. Analyse user requirements and prepare software requirements specifications.

CO7. Apply design principles of UML for software design.

CO8. Apply tools for developing UML diagrams.

CO9. Use cost estimation models for project evaluation.

CO10.   Work effectively as an individual to design UML models.

CO11.   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 plays 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.

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

II B. Tech. – I Semester
(20BT31232) PROGRAMMING WITH PYTHON LAB
(Skill Oriented Course)

(Common to CSSE, IT & CSBS)

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 practice on Python programming fundamentals, control structures, Sequences, Modular programming, Data representation and Visualization.

COURSE OBJECTIVES:

- To demonstrate practical knowledge on python programming constructs.
- To develop solutions for societal problems using python programming.

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

CO1. Demonstrate the practical knowledge on python programming constructs.

CO2. Apply python programming constructs to provide solutions for logical and societal problems.

CO3. Use python modules for data representation and visualization.

CO4. Work independently or in teams to solve problems with effective communication.

LIST OF EXPERIMENTS:

> ***Introduction to Python:*** *Introduction* **-** *tokens, literals, identifiers, keywords, special symbols and operators, fundamental data types, expressions, type conversions, Input and output, Selection Statements, Iterative Statements.*

1) a) Write a python program to perform the arithmetic operations on two integer numbers.

   b) Write a python program to evaluate the following expressions by reading necessary inputs from the keyboard.

   i) $ax^2 + bx + c$ ii) $ax^4 + bx^2 + c$ iii) $(ax - b) / (ax * b)$ iv) be^{kt}

2) a) Write a python program to swap two variable values without using a third variable.

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

   c) Write a python program to find roots of a Quadratic equation.

3) a) Write a python program that computes amount payable after discount from the price and quantity. The discount on amount is as follows:
   i) No discount when the amount less than or equal to 2000.

ii) 5% discount when the amount exceeds 2000.
iii) 10% discount when the amount exceeds 5000.
iv) 15% discount when the amount exceeds 10000.

b) Write a python program to find he largest among three numbers using nested if else.

c) Write a python program to print the following pattern.

```
            1
        1   2   1
    1   2   3   2   1
        1   2   1
            1
```

4) a) Write a python program to check the given number is a Fibonacci number or not.

b) Write a python program to find Armstrong numbers between *N1* and *N2*, where *N1* and *N2* are two integer numbers and *N1<N2*.

c) Write a python program to sort *n* numbers without using a predefined function.

Sequences*: Lists - operations, comprehensions, nested lists, tuples, nested tuples, strings - operations, string handling methods, string formatting, dictionaries - operations, diction   ary methods, sorting elements using lambdas.*

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

c) Write a python program to count the occurrences of each word in a given string.

6) Write a python program to implement the following conversions:

i) Decimal to Binary   ii) Binary to Decimal

7) a) Write a python program to read *n* number of student details into nested list and convert that as a nested dictionary.

b) Write a python program to generate a dictionary that contains numbers between 1 and *n* in the form (x, x+x).

Modular Programming*: Introduction to functions, scope and lifetime, return statement, positional, keyword, default arguments and variable-length arguments, recursive functions, NumPy module.*

8) a) Write a python function to perform the sum of integers that are sent to a function as variable length arguments.

b) Write a python program to implement the following using recursion:

i) GCD of two numbers   ii) Towers of Hanoi problem.

9) a) Write a python program to perform arithmetic operations on NumPy arrays.

b) Write a python program to perform following matrix operations using NumPy.

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

10) Write a python program that works as a scientific calculator using functions.

Data Representation and Visualization: *Pandas - data frame creation, reading data from CSV files, indexing and selecting data, dealing with rows and columns,*

Visualization - bar plots, histogram, Scatter Plot.

11) a) Write a python program to create Pandas DataFrame using list of lists.

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

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

   b) Draw histograms by considering an appropriate data set.

REFERENCE 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.

3. Reema Thareja, *Python Programming using Problem Solving Approach*, 1st Edition, Oxford University Press, 2017.

4. Charles Dierbach, *Introduction to Computer Science using Python: A Computational Problem-Solving Focus*, Wiley India, 2016.

III B. Tech. – I Semester

(20BT50532) **PYTHON FOR DATA SCIENCE LAB**

(Skill Oriented Course)

(Common to CSE & IT)

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

PRE-REQUISITES: A Course on "Programming with Python 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 OBJECTIVES:

- To impart knowledge on data manipulation and exploratory data analysis concepts that are vital for data science.
- To develop skills for applying python tools and techniques to analyze, visualize and interpret data.

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
    a) 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.

Software/Tools used:

- Python 3.8
- Python Libraries – NumPy, Pandas, Matplotlib, Seaborn, Beautiful Soup, Vader
- Anaconda Framework

ADDITIONAL LEARNING RESOURCES:

- https://swayam.gov.in/nd1_noc19_cs60/preview
- https://towardsdatascience.com/
- https://www.w3schools.com/datascience/
- https://github.com/jakevdp/PythonDataScienceHandbook
- https://www.kaggle.com

III B. Tech. – II Semester

(20BT61204) WEB DEVELOPMENT USING BOOTSTRAP AND ANGULAR JS

(Professional Elective-2)

(IT)

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

PRE-REQUISITES: Courses on Web Technologies.

COURSE DESCRIPTION: Advanced programming concepts of HTML5, CSS and Bootstrap; Angular JS; Directives & Building Databases; JSON and MongoDB.

COURSE OBJECTIVES:

- To impart programming skills on HTML5, CSS and Bootstrap.
- To develop Web applications using Angular JS,Directives & Building Databases, JSON and MongoDB.

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

CO1. Apply the HTML5, CSS3 and Bootstrap concepts in front-end development of modern web applications.

CO2. Design Web applications using Bootstrap.

CO3. Create and deploy scalable web-based system using Angular JS.

CO4. Implement Directives and Controllers for front-end development.

CO5. Demonstrate knowledge on the usage of Keys and Values Create Forms, validate and use Filters.

DETAILED SYLLABUS:

UNIT 1- HTML5 & CSS3   (9 periods)

HTML5 – Introduction, Elements, Tags, Lists, Tables, Images, Forms - Form Elements & Attributes,Hidden Fields, Semantic Elements, Media Elements, Canvas, SVG, Drag & Drop, Geolocation, WebStorage, Special Tags, Formatting Tags. CSS – Introduction, Styling, Box Model, Padding & Dimension, Transforms, Transitions, Animations, Multiple columns, User Interface

UNIT 2- BOOTSTRAP   (9 periods)

Bootstrap: Overview, Environment setup, Precompiled Bootstrap, Source Code, Grid System,Bootstrap CSS Overview, Typography, Code, Tables, Forms, Helper Classes, Responsive Utilities,Glyphicons, Dropdowns, Navigation Elements, Breadcrumb, Pagination, Badges, Progress bars

Plugins - Overview: Transition Plugin, Model Plugin, Dropdown Plugin, Scrollspy Plugin, Tab Plugin, Tooltip Plugin, Popover Plugin, Alert Plugin, Button Plugin, Collapse Plugin, Carousel Plugin, Affix Plugin

UNIT 3: INTRODUCTION TO ANGULAR JS   (9 periods)

Introduction: Features , Angular JSModel, View-Controller; Expression,Directives and Controllers; Angular JS Modules, Arrays,Working with ng-model, Working with Forms, Form Validation, Error Handling with Forms, Nested Forms with ng-form, Other Form Controls.

UNIT 4: DIRECTIVES & BUILDING DATABASES   (9 periods)

Filters: Using Filters in Controllers and Services; Angular JS Services, Internal Angular JSServices, Custom Angular JS Services, Directives, Alternatives to Custom Directives, Understanding the Basic options, Interacting with Server, HTTP Services, Building Database, FrontEnd and BackEnd.

UNIT 5: JSON AND MoNGoDB

JSON and MongoDB, Adopting a Non-relational Approach, Opting for Performance vs. Features Running the Database Anywhere, Generating or Creating a Key, Using Keys and Values, Implementing Collections

Total Periods: 45

- Topics for Self study are provided in the Lesson Plan.

TEXT BOOKS:

1. Shyam Seshadri, Brad Green, *AngularJS: Up and Running: Enhanced Productivity with Structured Web Apps*, Apress, First Edition, O'Reilly Media, Inc. September 2014.

2. Jon Duckett, *Web Design with HTML, CSS, JavaScript and jQuery Set Paperback*, 2014.

REFERENCE BOOKS:

1. Kristina Chodorow and Michael Dirolf, *Mongo DB – The Definitive Guide*, 1st Edition, O'Reilly Media, Inc. September 2010.

2. Jake Spurlock, *Responsive Web Development – Bootstrap*, 1st Edition, O'Reilly Media, Inc. May 2013.

III B. Tech. – II Semester

(20BT61232) **AI TOOLS LAB**

(IT)

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

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

COURSE DESCRIPTION: Speech Recognition, Language Detection and Translation, Object Detection, Human Body Estimation, Face Recognition, Optical Character Recognition, Chatbot.

COURSE OBJECTIVES:

- To impart knowledge on speech recognition, Language detection and Translation.
- To develop skills for 2D and 3D object detection, facial expression and text extraction.
- To provide skills to build Chatbot applications.
- To inculcate attitude to solve societal problems using AI Tools.

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

CO1: Demonstrate skills on speech recognition, Language detection and Translation and 2D and 3D object detection.

CO2: Use AI tools for text retrieval, facial expression detection and recognition.

CO3: Build Chatbot applications

CO4: Work independently or in teams to solve problems with effective communication skills.

LIST OF EXERCISES:

1. **Speech Recognition:** Read an audio file with Python and use the Google speech recognition API to perform conversion of Speech to Text.
2. Detect the language of the text using Language detection library (langdetect) ported from Google's language-detection.
3. **Language Translation:** Perform translation of given input text phrases or document using Google Translate API.
4. **Object Detection:** Detect multiple objects present in an image using Detectron-Facebook's free API.

5. **Human Body Estimation:** Extract a 3D mesh model of a human body from 2D RGB images using Facebook's DensePose tool.
6. Real time Smile Detection in Human Face using OpenCV.
7. Face Recognition using Microsoft Face API.
8. Test and evaluate the performance of the Google Tesseract OCR tool in recognizing printed Text characters.
9. **Chatbot:** Build a Chatbot capable of communicating and interacting with the users based on the given text input.

REFERENCE BOOKS:

1. Denis Rothman, *Artificial Intelligence by Example*, Packt Publishers, 2018.
2. Dr. Joshua Eckroth, *Python Artificial Intelligence projects for Beginners*, Packt Publishers, 2018.
3. Giuseppe Bonaccorso, Armando Fandango and Rajalingappaa Shanmugamani, *Python Advanced Guide to Artificial Intelligence*, Packt Publishers, 2018.
4. Dr. Shaila D Apte, Speech and Audio Processing, Wiley India.

III B. Tech. – II Semester

(20BT61233) ADVANCED WEB DEVELOPMENT LAB

(Skill Oriented Course)

(IT)

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

PRE-REQUISITES: A Course on Web Technologies.

COURSE DESCRIPTION: Node.js framework, React JS, Django framework, PyCharm Integrated Development Environment.

COURSE OBJECTIVES:

- To provide hands-on experience on Web application Development using Node.js framework, React JS, Django framework, PyCharm Integrated Development Environment.

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

CO1: Develop front-end applications using Node.js framework and React JS.

CO2: Develop server-side Framework using Django.

CO3: Building web application and Host web application using front-end and back-end tools.

CO4: Work independently or in teams to solve problems with effective communication.

LIST OF EXERCISES:

1. Front-end Web Application Library	
State-of-the-art Front-End Library: **React** **Pre-requisites:** HTML and JavaScript.	Experiments: 1. Installing Node.js framework and configuring Visual Studio (VS) Code Integrated Development Environment (IDE), and its dependencies. 2. Create and Run "Hello World" Application in VS Code. 3. Create a React application that includes simple functional components. 4. Create a React application that includes simple class components.

	5. Develop a React application to insert and access props (properties) and state of components.
	6. Create a React application to demonstrate event handling.
	7. Develop a React application for list rendering.
	8. Implement a React application for form handling.

2: Server-side Development Framework.

State-of-the-art server-side Framework: **Django** **Pre-requisites:** Python	Experiments: 1. Installing Python, Django framework and configuring PyCharm Integrated Development Environment (IDE), and its dependencies. 2. Creating workspace, project and setting up the necessary environment. 3. Implement a simple view to handle http response (display Hello World) in Django Application. 4. Create a simple model for storing student details. 5. Implement a Django application for form creation and storage of form data into model. 6. Write simple test cases and test any Django application. 7. Create a Django application to include static files such as images, CSS and JavaScript.

3. Hosting Web Applications

Building web application and Hosting web application	Experiments: 1. Choosing a hosting server and selecting a plan for web hosting. 2. Choosing and configuring DNS address 3. Uploading, configuring and running the website over the internet.

REFERENCE BOOKS:

1. Cory Gackenheimer, *Introduction to React*, Apress, 2015.
2. Ethan Brown, *Web Development with Node & Express*, O'Reilly, 2014

Nigel George, *Mastering Django: Core*, Packt, 2016

IV B.Tech. – I Semester

(20BT71209) DIGITAL AND SOCIAL MEDIA MARKETING

(Professional Elective-5)

(IT)

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

PRE-REQUISITES: A course on Computer Networks.

COURSE DESCRIPTION:

Digital marketing strategies, different types of social media marketing tools, understanding customer journeys, Business Model for e-commerce, structure a digital marketing strategy, Setting SMART objectives, The Operations and Marketing Connection, Social CRM and Blogger Outreach, Engagement as a Business Activity.

COURSE OBJECTIVES:

- To provide knowledge on digital marketing strategies and applications.
- To study social media technologies and customer management issues.

COURSE OUTCOMES:

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

CO1. Demonstrate knowledge on digital Marketing Strategies.

CO2. Analyze the marketing potential of digital technologies and social media platforms for a particular real-life marketing challenge.

CO3: Analyze digital marketing strategies for improving digital marketing.

CO4: Identify the Scope of Social Interaction, Customer Relationships.

CO5. Design social business Techniques for business analysis.

DETAILED SYLLABUS:

UNIT-1: INTRODUCTION TO DIGITAL MARKETING   (9 Periods)

Digital Marketing Fundamentals, Key features of digital marketing strategies, Applications of Digital Marketing, Benefits of Digital marketing, Alternative digital business models, The relationship between digital and traditional communications, different types of social media marketing tools, Key communications concepts for digital marketing.

UNIT-2: ONLINE MARKETPLACE ANALYSIS   (9 Periods)

Situation analysis for digital marketing, Digital marketing environment, Understanding customer journeys, Consumer behaviour and implications for Marketing, Competitors, Suppliers, Business Model for e-commerce.

UNIT-3: DIGITAL MARKETING STRATEGY   (9 Periods)

The need for an integrated digital marketing strategy, How to structure a digital marketing strategy,Situation analysis,Setting goals and objectives for digital marketing, Strategy formulation for digital marketing,The need for integrated digital marketing , strategy implementation. Setting SMART objectives.

UNIT-4: SOCIAL MEDIA AND CUSTOMER ENGAGEMENT   (9 Periods)

The Social Feedback Cycle,The Social Web and Engagement,The Operations and Marketing Connection,The New Role: Social Interaction,Customer Relationships: CRM Gets Social,Outreach and Influencer Relations,Social CRM and Blogger Outreach,Build a Social Business.The Social Business Ecosystem.

UNIT-5: SOCIAL TECHNOLOGY AND BUSINESS DECISIONS   (9 Periods)

Create a Social Business,Understand the Conversations That Matter,Social CRM and Decision Support,Social Analytics,Know Your Influencers,Engagement on the Social Web,Engagement as a Customer Activity,Engagement as a Business Activity,Social CRM and Business Design.

TEXT BOOKS:

1. Dave chaffey and Fiona ellis-chadwick ,Digital Marketing strategy, implementation and, 6th Edition, 2016.

2 .Dave Evans ,Social Media Marketing: The Next Generation of Business Engagement, Wiley Piblishing.Inc, 2010

REFERENCE BOOKS:

1. Ryan, Damian; Understanding Digital Marketing: marketing strategies for engaging the digital generation; Kogan Page , 3rd Edition, 2014.

2. MoutsyMaiti: Internet Mareting, Oxford University Press India

3. Eric Greenberg, and Kates, Alexander; Strategic Digital Marketing: Top Digital Experts Share the Formula for Tangible Returns on Your Marketing Investment; McGraw-Hill Professional, 2013.

IV B. Tech. - I Semester

(20BT71233) AUGMENTED REALITY AND VIRTUAL REALITY APPLICATIONS LAB

(Skill Oriented Course)

(IT)

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

PRE-REQUISITES: A course on object-oriented programming.

COURSE DESCRIPTION: Hands-on experience on Unity, visual studio installation, Creating and Using Scripts Controlling Game Objects using components, Creating 2D platform game, Player Moving, Platform Moving, Creating 3D game, Face Tracking and location-based augmented reality application.

COURSE OBJECTIVES:

1. To introduce the importance and applications of Augmented Reality and Virtual Reality Systems
2. To develop creativity and individuality in problem solving and performing tasks.
3. To learn the basic algorithms and programming techniques of 3D game development using C#

COURSE OUTCOMES:

On successful completion of this course, the student will be able to:

CO1: Demonstrate skills needed to implement all of the necessary assets for a game such as music, sound, levels, programming, design.

CO2: Analyze the requirements to develop a new game.

CO3: Develop 3D virtual environments, and 3D interaction techniques for game development.

CO4: Develop immersive virtual reality applications for societal needs.

CO5: Work individually or in teams to solve problems with effective communications.

List of Experiments:

1. Installation of Unity and Visual Studio, setting up Unity for VR development, understanding documentation of the same.

2. Demonstration of the working of HTC Vive, Google Cardboard, Google Daydream and Samsung gear VR.
3. Develop a scene in Unity that includes:
    i.   A cube, plane and sphere, apply transformations on the 3 game objects.
    ii. Add a video and audio source.
4. Develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the color, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the color and material/texture of the game objects dynamically on button click.
5. Develop a scene in Unity that includes a sphere and plane. Apply Rigid body component, material and Box collider to the game Objects. Write a C# program to grab and throw the sphere using VR controller.
6. Develop a simple UI (User interface) menu with images, canvas, sprites and button. Write a C# program to interact with UI menu through VR trigger button such that on each successful trigger interaction display a score on scene.
7. Create an immersive environment (living room/ battlefield/ tennis court) with only static game objects. 3D game objects can be created using Blender or use available 3D models.
8. Include animation and interaction in the immersive environment (living room/ battlefield/ tennis court).
9. Create a 2D platform game in unity. Add the Bullets, Add animations to Player and Platforms, Add Coins, Score, audio and health bar and create Enemy, also different levels.
10. Create a 3D game Infinite Runner in unity, and the project structure should include Materials, Prefabs, Scripts, scene etc. Create Collectable Objects and Collecting the Pick Up Objects. Display the Score and Health bar.
11. Create an application to detect faces in a front-camera AR experience, overlay virtual content, and animate facial expressions in real-time using a front depth camera.
12. Create a location-based augmented reality application to navigate travelers through a large hotel or airport and help to learn more about popular sightseeing spots nearby.

REFERENCE BOOKS:

1. Erin Pangilinan, Steve Lukas, and Vasanth Mohan, *Creating Augmented and Virtual Realities Theory and Practice for Next-Generation Spatial Computing*, O'Reilly Media, 2019.
2. Jeff W. Murra, *Building Virtual Reality with Unity and SteamVR,* CRC Press, 2017
3. Penny de Byl, *Holistic Game Development with Unity An All-in-One Guide to Implementing Game Mechanics, Art, Design and Programming*, 2nd Edition, Taylor & Francis Group, LLC, 2017
4. Vahé Karamian, *Introduction to Game Programming: Using C# and Unity 3D*, Noorcon Inc. 2016.
5. JOSEPH HOCKING, *Unity in Action Multiplatform game development in C#*, 2nd Edition, MANNING Shelter Island, 2018.

IV B. Tech. – I Semester

(20BT712AC) DEEP LEARNING TECHNOLOGIES

(Audit Course)

(IT)

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

PRE-REQUISITES: A Course on Data Analytics.

COURSE DESCRIPTION: Introduction to Deep Learning, *Keras* and *Tensorflow*; Working with Keras; Deep Learning for Computer Vision; Deep Learning for Timeseries and Text; Generative Deep Learning and Best Practices.

COURSE OBJECTIVES:

- To impart knowledge on the concepts of deep learning.
- To understand the deep learning models to solve the problems of computer vision.
- To build deep learning models using *Keras* and *TensorFlow* to solve societal problems.

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

CO1. Demonstrate basic knowledge on deep learning models.
CO2. Build and evaluate machine and deep learning models using *Keras* and *TensorFlow*.
CO3. Develop convolutional neural networks (*convnets*) models to solve image recognition and classification problems.
CO4. Apply recurrent neural networks for speech recognition and natural language processing.
CO5. Build generative deep learning models to find new data instances for classification.

DETAILED SYLLABUS:

UNIT I – INTRODUCTION TO DEEP LEARNING   (6 periods)

Introduction: Artificial Intelligence, Machine Learning and Deep Learning, a brief history of machine learning, mathematical building blocks of neural networks - a first look at a neural network, data representations, tensor operations, gradient-based optimization.

UNIT II – KERAS AND TENSORFLOW   (6 periods)

Introduction: Setting up a deep learning workspace, first steps with TensorFlow, understanding core Keras API, classifying movie reviews, classifying newswires, predicting house prices, evaluating machine-learning models, improving model fit.

Working with Keras: A spectrum of workflows, Different ways to build Keras models, using built-in training and evaluation loops, writing your own training and evaluation loops.

UNIT III – DEEP LEARNING FOR COMPUTER VISION   (6 periods)

CONVNETS: Introduction to convolutional neural networks (*convnets*), training a *convnet* from a scratch, leveraging a pre-trained model, three essential computer vision tasks, image segmentation example, modern *convnet* architecture patterns, interpreting what *convnets* learn.

UNIT IV – DEEP LEARNING FOR TIMESERIES AND TEXT   (6 periods)

Timeseries: Different kinds of timeseries tasks, temperature forecasting example, Understanding Recurrent Neural Networks (RNN), advanced use of RNN.

Text: Natural Language Processing - A bird's eye view, preparing text data, sets and sequences, transformer architecture, sequence-to-sequence learning.

UNIT V – GENERATIVE DEEP LEARNING AND BEST PRACTICES   (6 periods)

Generative Deep Learning: Text generation, DeepDream, Neural style transfer, generating images with variational auto-encoders, introduction to generative adversarial networks.

Best practices: Getting the most our of your models, scaling-up model training.

Total Periods: 30

- Topics for Self study are provided in the Lesson Plan.

TEXT BOOKS:

1. François Chollet, "*Deep Learning with Python*", Manning Publications, 2nd Edition, 2021.

REFERENCE BOOKS:

1. Charu C. Aggarwal , "*Neural Networks and Deep Learning*", Springer; 1st Edition, 2018.
2. Sudharsan Ravichandiran, "*Hands-On Deep Learning Algorithms with Python*", Packt Publishing, 2019.
3. Josh Patterson and Adam Gibson, "*Deep Learning - A Practitioner's Approach*" O'Reilly Media, 1st Edition, 2017.
4. Ian Goodfellow, Yoshua Bengio and Aaron Courville, "*Deep Learning - Adaptive Computation and Machine Learning series*", MIT Press, 2016.