



Progressive Education Society's

**Modern College Of Arts, Science and
Commerce, Ganeshkhind, Pune - 411
016**

(Revised NEP Curricular framework)

4th May 2024.

Syllabus for

F.Y.B.B.A(CA)

A.Y 2024-2025

FYBBA(CA)

Subject List

FYBBA(CA) Sem I

Course Type	Sr . No.	Course (Subject) Title	Course (Subject) code	Credits	Weightage for Internal Marks	Weightage For External Marks	Weightage for practical	Total Marks
DSC 1	1	Introduction to Python Programming (2T)	24BBA11101	2	20	30	50	100
		Computer Laboratory Based on Python Programming (2P)	24BBA11102	2				
DSC 2	2	Database Management System(2T)	24BBA11103	2	20	30	50	100
		Computer Laboratory Based on DBMS (2P)	24BBA11104	2				
DSC 3	3	Business Communication (T)	24BBA11105	4	70	30		100
OE1	4	OE1 from Arts /Science Basket	-	2	20	30		50
SEC1	5	Design Thinking	24BBA11406	2	20	30		50
AEC1	6	English / Foreign language	-	2	20	30		50
VEC1	7	Value Education (T)	-	2	20	30		50
IKS Gen	8	Foundation Course on Indian knowledge system I	24IKS11501	2	20	30		50

FYBBA(CA) Sem II

Course Type	Sr . No.	Course(Subject) Title	Course(Subject) code	Credits	Weightage for Internal Marks	Weightage for External Marks	Weightage for practical	Total Marks
DSC 1	1	Advanced Python Programming (2T)	24BBA12101	2	20	30	50	100
		Computer Laboratory Based on Advanced Python Programming(2P)	24BBA12102	2				
DSC 2	2	Relational Database Management System(2T)	24BBA12103	2	20	30	50	100
		Computer Laboratory Based on RDBMS(2P)	24BBA12104	2				
DSC 3	3	Entrepreneurship Essentials(T)	24BBA12105	4	70	30		100
OE2	4	OE2 from Arts / Science Basket		2	20	30		50
SEC2	5	Web Technologies	24BBA12406	2	20	30		50
AEC2	6	English/Foreign Language		2	20	30		50
VEC2	7	Democracy Election and Governance(T)		2	20	30		50
CC2	8	CC1 from College Basket		2	20	30		50
			Total Credits SEM I (22cr)+SEM II (22cr)	44 Cr				

SEM I

Subject Code: 24BBA11101

Subject: Introduction to Python Programming (2T)

Total Hours :- 30

Total Credits: 02

Course Objectives:

1. Student will learn and understand Python programming basics and paradigm.
2. Student To learn and know the concepts of file handling, exception handling.

Course Outcomes:

 On completion of the course,

1. Student will be able define and demonstrate the use of built-in data structures “lists” and “dictionary”.
2. Student will be able Design a program to solve a real world problem.

Unit	Details	Lectures
I	Unit 1: Introduction to Python 1.1 History, feature of Python, setting up path, working with python Interpreter, basic syntax, variable and data types, operators 1.2 Conditional statements -If, If-Else, nested if-else, Examples. 1.3 Looping -For, While, Nested loops, Examples 1.4 Control Statements -Break, Continue, Pass. 1.5 String Manipulation -Accessing String, Basic Operations, String Slices, Function and Methods, Examples. 1.6 Lists -Introduction, accessing list, operations, working with lists, function & methods. 1.7 Tuple -Introduction, accessing tuples, operations working, function & methods, Examples. 1.8 Dictionaries -Introduction, accessing values in dictionaries, working with dictionaries, properties, function, Examples. 1.9 Functions -Defining a function, calling a function, types of function, function arguments, anonymous function, global & local variable, Examples.	10
II	Unit 2: Modules and Packages 2.1 Built in Modules 2.1.1 Importing modules in python program 2.1.2 Working with Random Modules. 2.1.3 E.g. - built-ins, time, date time, calendar, sys, etc 2.2 User Defined functions 2.2.1 Structure of Python Modules 2.3 Packages 2.3.1 Predefined Packages 2.3.1 User defined Packages	8
III	Unit 3: Classes, Objects 3.1 Classes and Objects 3.1.1 Classes as User Defined Data Type 3.1.2 Objects as Instances of Classes 3.1.3 Creating Class and Objects 3.1.4 Creating Objects by Passing Values 3.1.5 Variables & Methods in a Class	12
	Total	30

Reference Books:

1. Mark Lutz, Programming Python, O`Reilly, 4th Edition, 2010
2. Dive into Python, Mike
3. Learning Python, 4th Edition by Mark Lutz
4. Programming Python, 4th Edition by Mark Lutz
5. Python Programming:An introduction to computer,John Zelle,3rd Edition.

Subject Code: 24BBA11102

Subject: Computer Laboratory Based on Python Programming (2P)

Total Hours :- 30

Total Credits: 02

Sr. No.	Assignment Name	Practicals
1	Introduction to Basic Python	5
2	Working with Strings and List	6
3	Working with Tuples, Sets and Dictionaries	6
4	Working with Functions, Modules and Packages	5
5	Python Classes and Objects	8
Total (Out of 30)		30
Total (Out of 5)		

Assignment on Basic Python

- 1) Write python script to calculate sum of digits of a given input number.
- 2) Write python script to check whether a input number is Armstrong number or not.
- 3) Write python script to check whether a input number is perfect number or not.
- 4) Write a program to calculate XY
- 5) Write a program to check whether a input number is palindrome or not.
- 6) Write a program to calculate sum of first and last digit of a number.
- 7) Write a program to accept a number and count number of even, odd, zero digits within that number.
- 8) Write a program to accept a binary number and convert it into decimal number.
- 9) Write a program which accepts an integer value as command line and print "Ok" if value is between 1 to 50 (both inclusive) otherwise it prints "Out of range"
- 10) Write a program which accept an integer value 'n' and display all prime numbers till 'n'.
- 11) Write python script to accept two numbers as range and display multiplication table of all numbers within that range.
- 12) Write a python script to generate the following pattern upto n lines
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
- 13) Write a python script to create a list and display the list element in reverse order
- 14) Write a python script to display alternate characters of string from both the direction.

- 15) Write a python program to count vowels and consonants in a string.
- 16) Write a python script which accepts 5 integer values and prints “DUPLICATES” if any of the values entered are duplicates otherwise it prints “ALL UNIQUE”. Example: Let 5 integers are (32, 45, 90, 45, 6) then output “DUPLICATES” to be printed.
- 17) Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

Sample String : 'abc', 'xyz'

Expected Result : 'xycabz'

- 18) Write a Python program to count the occurrences of each word in a given sentence.
- 19) Write a Python program to add and remove operation on set.
- 20) Write a Python program to do iteration over sets.
- 21) Write a Python program to find the length of a set.
- 22) Write a Python program to create a tuple with numbers and print one item.
- 23) Write a Python program to add an item in a tuple.
- 24) Write a recursive function which print string in reverse order.
- 25) Write a python script using function to calculate XY.
- 26) Write a python script to generate Fibonacci terms using generator function.
- 27) Write a Python Program to Accept, Delete and Display students details such as Roll.No, Name, Marks in three subject, using Classes. Also display percentage of each student.
- 28) Write Python class to perform addition of two complex numbers using binary + operator overloading.
- 29) Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area and volume.
- 30) Python Program to Create a Class which Performs Basic Calculator Operations.

Total Contact Hours: -30

Total Credits: - 2

Subject Code: - 24BBA11103

Subject Name -: Database Management System(2T)

Objective :

- i) Enables students to understand database concepts.
- ii) To understand advantages of DBMS over File Processing System.
- iii) Integrity of data means to bring the accuracy and consistency.

Outcomes:

- i) CO1: Able to understand basic database concepts in database system.
- ii) CO2: To get knowledge of Front End and Backend.
- ii) CO3: Able to write SQL queries and do database connectivity with any front-end platform

Chapter No.	Name of Chapter and Contents	Lectures
1	<p>File Structure and Organization</p> <p>1.1 Introduction</p> <p>1.1.1 File</p> <p>1.1.2 Logical and Physical Files Definitions</p> <p>1.3 Basic File Operations</p> <p>1.3.1 Opening Files</p> <p>1.3.2 Closing Files</p> <p>1.3.3 Reading and Writing</p> <p>1.3.4 Seeking</p> <p>1.4 File Organization</p> <p>1.4.1 Field and Record structure in file</p> <p>1.4.2 Record Types</p> <p>1.4.3 Introduction to file organization</p>	3
2	<p>Database Management System</p> <p>2.1 Introduction</p> <p>2.2 Basic Concept and Definitions</p> <p>2.2.1 Data Vs Information</p> <p>2.2.2 Data Dictionary</p> <p>2.3 Definition of DBMS</p> <p>2.4 Applications of DBMS</p> <p>2.5 File processing system Vs. DBMS</p> <p>2.6 Advantages and Disadvantages of DBMS</p> <p>2.7 Users of DBMS</p> <p>2.7.1 Database Designers</p> <p>2.7.2 Application programmer</p> <p>2.7.3 Sophisticated Users</p> <p>2.7.4 End Users</p> <p>2.8 Views of Data</p> <p>2.9 Data Models</p> <p>2.9.1 Object Based Logical Model</p> <ul style="list-style-type: none"> • Object Oriented Data Model • Entity Relationship Data Model <p>2.9.2 Record Base Logical Model</p> <ul style="list-style-type: none"> • Relational Model • Network Model • Hierarchical Model <p>2.10 Entity Relationship Diagram(ERD)</p> <p>2.11 Extended features of ERD Overall System structure</p>	6
3	<p>Relational Model</p> <p>3.1 Introduction</p> <p>3.2 Terms</p> <p>a. Relation</p> <p>b. Tuple</p>	6

	c. Attribute d. Cardinality e. Degree of relationship set f. Domain 3.3 Keys 3.3.1 Super Key 3.3.2 Candidate Key 3.3.3 Primary Key 3.3.4 Foreign Key	
4	SQL (Structured Query Language) 4.1 Introduction 4.2 DDL Commands 4.3 DML Commands 4.4 Constraints in SQL. 4.5 SQL Functions-Date, Time, Numeric, String ,Conversion functions 4.6 Simple Queries 4.7 Nested Queries 4.10 Aggregate Functions	15
	Total	30

References:

- 1) Database System Concepts By Henry korth and A. Silberschatz
- 2) SQL, PL/SQL The Programming Language Oracle :- Ivan Bayross, BPB Publication.
- 3) Database Systems Concepts, Designs and Application by Shio Kumar Singh, Pearson
- 4) Introduction to SQL by Reck F. van der Lans by Pearson
- 5) Modern Database Management by Jeffery A Hoffer , V.Ramesh, Heikki Topi ,Pearson
- 6) Database Management Systems by Debabrata Sahoo ,Tata MacgrawHill

Total Contact Hours: -30

Total Credits: - 2

Subject Code: - 24BBA11104

Subject Name -: Computer lab based on Database Management System(2P)

Topics for Laboratory Assignments for DBMS	
Sr. No	Assignment
1	DDL(Data Definition Language)
2	Alter Table and Drop Table
3	DML Commands
4	RDB without Constraints
5	Table Creation with Constraints
6	RDB with Constraints
7	Demonstration of Select Command
8	SQL Set operations
9	Joins

Sr. No	Assignment
1	<p>Creating Tables:</p> <p>❖ Create table for the information given below by choosing appropriate data types and also specifying proper primary key constraint on fields which are underlined</p> <ol style="list-style-type: none"> 1. Player (<u>player_id</u> , name, Birth_date ,Birth_place, game_name) 2. Student (roll_no, name,class,per,birth_date) 3. Project (project_id, project_name , project_description ,status) 4. Donor (donor_no, donor_name,blood_group,last_date) <p>❖ Create table for the information given below by choosing appropriate data types and also specifying proper primary key constraint on fields which are underlined.</p> <ol style="list-style-type: none"> 1. Property (<u>property_id</u>, property_desc , area, rate, agri_status) 2. Actor (<u>actor_id</u>, Actor_name, birth_date) 3. Movie(movie-no, name, release-year) 4. Hospital(hno,hname,hcity) <p>❖ Create table for the information given below by choosing appropriate data types and also specifying proper primary key constraint on fields which are underlined.</p> <ol style="list-style-type: none"> 1. Employee(ENo, EName, <u>Joining_date</u>,company_name,salary,Designation) 2. College(College_Code,College_Name,Address,Establish_year) 3. Doctor(Dno, Dname, Specialization,Qualification) 4. Classroom(CRoomNo,location,capacity)

2	<p>Create table student(Roll_no, sname, date_of_birth). Add new column into student relation named address as a text data type and a column phone of data type integer.</p> <ul style="list-style-type: none"> ❖ Create table driver (licence_no, Name, Address) and perform the following queries <ol style="list-style-type: none"> 1. Add new column age of data type integer. 2. Alter table by modifying driver_name to "Patil" 3. Alter table driver ,drop the column age. 4. Remove the driver table from the database. ❖ Create table Game (name, no-of-players, captain_name) and perform the following queries <ol style="list-style-type: none"> 1. Add new column game_no of data type integer. 2. Alter table by adding constraint uppercase to captain_name. 3. Modify table by adding the column game_duration. 4. Add column game_type with values cricket,hockey,tennis. 5. Remove game table from the database.
3	<ul style="list-style-type: none"> ❖ Consider the following table Employee(ENo, EName, Salary, DOJ,Qualification) and answer the following query. <ol style="list-style-type: none"> 1. Insert at least five records into the table. 2. Update the salary of employee to 50000 whose ENo is 1. 3. Delete the details of employee whose ENo is 5. 4. Update the Qualification of employee to "MCS NET" whose Name is Mr.Satyavan. 5. Update the salary of employee to 40000 whose qualification is "MCS NET" and Name is "Ajay" ❖ Consider the following table Hospital (HNo, HName, Addr, Est_Year , speciality) and answer the following query. <ol style="list-style-type: none"> 1. Insert at least five records into the table. 2. Update an address of hospital to "Pimple Gurav" whose name is "Birla". 3. Update the specialty of hospital to "Multi" whose established year is between 1990 to 2000. 4. Delete the details of Hospital whose address is "Pimpri". ❖ Consider the following table Student (Roll_No, Name, class, DOB, college) and answer the following query. <ol style="list-style-type: none"> 1. Insert at least 10 records into the table. 2. Update the class of student to "TY" whose birth date is '18/03/1999'. 3. Delete the details of students whose college is "Dr.D Y Patil". 4. Update the college of student to "Dr. D Y Patil " whose name is "Yash".
4	<ul style="list-style-type: none"> ❖ Consider the following entities and their relationships. Create a RDB in 3 NF for the following and answer the queries: Emp(eno ,ename ,designation ,salary,DOJ) Dept(dno,dname ,loc) The relationship between Dept & Emp is one-to-many. <ol style="list-style-type: none"> 1. Insert at least five records into the tables. 2. Display the names of employees who are working in "Quality Department". 3. Display the name of employee who is 'Manager' of "Purchase Department". 4. Display the name of department whose location is "Baramati" and "Mr. Pawar" is

	<p>working in it.</p> <p>5. Display the names of employees whose salary is greater than 50000 and department is "Quality".</p> <p>❖ Consider the following entities and their relationships. Create a RDB in 3 NF for the following and answer the queries: Hospital(hno ,hname , city, Est_year) Doctor(dno , dname , addr, Speciality) The relationship between Hospital and Doctor is one - to – Many</p> <ol style="list-style-type: none"> 1. Insert at least 10 records into the tables. 2. Display the names of hospitals which are located at "Pimpri" city. 3. Display the names of doctors who are working in "Birla" Hospital and city name is "Chinchwad". 4. Display the specialty and name of doctor who is working in "Ruby" hospital and his address is "Pimple Gurav". 5. Display the names of doctors whose speciality is "medicine" <p>❖ Consider the following entities and their relationships. Create a RDB in 3 NF for the following and answer the queries: Patient (PCode, PName , Addr , Disease) Bed (Bed_No, RoomNo, loc) Relationship: - A one-one relationship between patient and bed.</p> <ol style="list-style-type: none"> 1. Insert at least five records into the tables. 2. Display the names of patients who are admitted in room no 101. 3. Display the disease of patient whose bed_No is 1. 4. Give the roon_no and bed_no of patient whose name is "Mr Ajay".
5	<p>□ Consider the following tables and integrity constraints given and create the tables accordingly:□</p> <p>1. Machine(Mid, MName NOT NULL, MType, MPrice , MCost) Constraints: 1. MName should be in uppercase. 2. MType can be ('drilling', 'milling', 'lathe', 'turning', 'grinding'). 3. MPrice should be greater than zero. Table level constraint: MCost less than MPrice.</p> <p>2. Policy(No, Name NOT NULL, Type , Sale_Date, Intro_date) Constraints: 1. Name should be in lowercase. 2. Type can be ('life', 'vehicle', 'accident')</p> <p>Table level constraint: Sale_date should be greater than Intro_date.</p> <p>3. Employee (EmpNo, Emp_Name NOT NULL, Emp_desig, Emp_sal , Emp_uid) Constraints: 1. Emp_name should be in uppercase. 2. Emp_desg can be ('Manager', 'staff', 'worker'). 3. Emp_sal should be greater than zero.</p>

	<p>Table level constraint: Emp_uid not equal to Emp_id</p> <p>3 Room(room_no , type, price);</p> <p>Constraints:</p> <ol style="list-style-type: none"> 1. Room type must be one of single, double, family. 2. Price must be between Rs.500/- and 1000/-. 3. Room no must be between 1 and 100.
6	<p><input type="checkbox"/> Consider the following Entities and Relationships</p> <p>Sales_order(ordNo, ordDate)<input type="checkbox"/></p> <p>Client (clientNo, ClientName, addr)</p> <p>Constraint: Primary key, ClientName should not be NULL.</p> <p>A client can give one or more sales_orders ,but a sales_order belongs to exactly one client. Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form(3 NF) and perform the following tasks.</p> <ol style="list-style-type: none"> 1. Insert two client records into client table. 2. Insert 3 sales records for each client. 3. Change order date of client_No 'C004' to 18/03/2019 4. Delete all sale records having order date before 10 /02/2018. 5. Display date wise sales_order given by clients. <p><input type="checkbox"/> Consider the following Entities and Relationships</p> <p>Customer (cust_no, cust_name, address, city)</p> <p>Loan (loan_no, loan_amt)</p> <p>Relation between Customer and Loan is Many to Many</p> <p>Constraint: Primary key, loan_amt should be > 0.</p> <p>Create a Database in 3NF & write queries for following.</p> <ol style="list-style-type: none"> 1. Find details of all customers whose loan is greater than 10 lakhs. 2. List all customers whose name starts with 'sa'. 3. List names of all customers in descending order who has taken a loan in Pimpri city. 4. Display customer details having maximum loan amount. 5. Calculate total of all loan amount. <p><input type="checkbox"/> Consider the following Entities and Relationships</p> <p>Department (dept_no, dept_name, location)</p> <p>Employee (emp_no, emp_name, address, salary, designation)</p> <p>Relation between Department and Employee is One to Many</p> <p>Constraint: Primary key, salary should be > 0.</p>

	<p>Create a Database in 3NF & write queries for following.</p> <ol style="list-style-type: none"> 1. Find total salary of all the employees from computer science dept. 2. Find the name of department whose average salary is above 10000. 3. Count the number of employees in each department. 4. Display the maximum salary of each department. 5. Display department wise employee list. 6. Increase Salary of “Managers” by 15% 7. Delete all Employees who are working as “clerk”.
7	<p>Create the following tables (primary keys are underlined). Emp(eno,ename,sal,address,ph_no) Dept(dno, name, loc) Emp and Dept are related with many to one with each other. Create the Relations accordingly, so that the relationship is handled properly and relations are in normalized form (3NF). Execute following select queries & write the business task performed by each query.</p> <ol style="list-style-type: none"> 1. Select * from emp; 2. Select empno, name from emp; 3. Select distinct deptno from emp; 4. Select * from emp where deptno = ; 5. Select * from emp where address = 'pune' and sal > ___; 6. Select * from emp where address = 'pune' and salary between _ and _____; 7. Select * from emp where name like '---%' 8. Select * from emp where name like '%and%' 9. Select * from emp where salary is null; 10. Select * from emp order by eno; 11. Select * from emp order by deptno, eno desc; 12. Select deptno as department, sum(salary) as total from emp group by deptno order by deptno; 13. Select deptno as department , count(eno) as total_emp from emp group by deptno having count(eno) > order by deptno; 14. select avg(salary) from emp; 15. select max(salary),deptno from emp group by deptno having max(sal) > _____; 16. select deptno, min(salary) from emp order by deptno; 17. update emp set salary = salary + 0.5*salary where deptno = (select deptno from department where dname = 'finance'); 18. update emp set deptno = (select deptno from department where dname = 'finance') Where deptno = (select deptno from department where dname = inventory');
8	<p>Create the following tables. (Primary Keys are underlined) Emp(emp_id ,emp_name, address, bdate) Investor (inv_name , inv_no, inv_date, inv_amt) An employee may invest in one or more investments; hence he can be an investor. But an investor need not be an employee of the firm. Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF). Assume appropriate data types for the attributes. Add any new attributes, as required</p>

by the Queries. Insert sufficient number of records in the relations / tables.

Write the following queries & execute them.

1. List the distinct names of customers who are either employees, or investors or both.
2. List the names of customers who are either employees, or investors or both.
3. List the names of employees who are also investors.
4. List the names of employees who are not investors.

❖ **Create the following tables. (Primary Keys are underlined)**

Student (rno,sname,address,class)

Subject (subno,subname)

Student and Subject are related with many-to-many relationship with attribute marks and status. Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

Write the following queries & execute them.

1. List the distinct names of students who have either Electronics, or Statistics or both subjects.
2. List the names of students who are either passed or failed.
3. List the students who have "Database" subject and they are not in "TY" class.
4. List the names of students who are not failed in any subject.
5. List the names of students not staying at "Uruli Kanchan".

9

❖ **Consider the following relations to understand the use of joins.**

Student (s_id , sname, level ,age , subject)

Class (cname , meetat ,room, fid)

Enrolled (s_id i, cname)

Faculty (fid ,fname ,deptid)

The meaning of above relationship is enrolled has one record per student _class pair such that the student is enrolled in the class. Read the query carefully and insert sufficient

number of records in the relations / tables with appropriate values to perform the following queries.

1. Find the names of all classes that either meet in room R128 or have five or more students enrolled.

Sql>Select c.name from class c where c.room ='r128' or c.name in (select e.cname from enrolled e group by e.name having count(*)>= 5);

2. Find the name of the oldest student who is either a history subject or enrolled in a course taught by I.teach.

Sql> Select max(s.age) from student s where (s.subject='history') or s.num in (select e.num from class c ,enrolled e ,faculty f where e.name =c.name and c.fid=f.id and f.fname ='I.teach);

3. Find the names of students enrolled in the maximum number of classes.

Sql> Select distinct s.name from student s where s.num in (select e.num from enrollede group by e.num having count(*) >=all (select count(*) from enrolled e2 group by e2.num));

4. Find the names of student not enrolled in any class.

Sql> Select distinct s.name from student s where s.num not in (select e.num from enrolled e);

5. Find the names of faculty members who teach in every room in which some class is taught.

Sql> Select distinct f.name from faculty f where not exists ((select * from class) except (select c1.room from class c1 where c1.fid = f.fid));

Subject Name -: Design Thinking

Subject code -: 24BBA11406

Total Contact Hours: -30

Total Credits: - 2

Pre requisite: Basic Mathematics

Objectives: To develop Analytical / Logical thinking and problem solving capabilities.

Course Outcomes:

CO1: Analytical and Logical Thinking is developed amongst students.

CO2: Able to find solution of problems using Problem Solving Techniques.

CO3: Learn Basic idea of programming.

CO4: Able to write their own algorithms.

Unit No.	Contents	No of hours
1	Flowchart 1.1 Introduction 1.2: Flowcharts (Definitions, Symbols) 1.3 Examples (Write algorithms and draw flowcharts) 1.3.1 Addition / Multiplication of integers. 1.3.2 Determining if a number is +ve / -ve / even / odd. 1.3.3 Maximum of 2 numbers, 3 numbers. 1.3.4 Sum of first n numbers, sum of given n numbers, Sum of digits of a given number, sum of first and last digit of a number. 1.3.5 Digit reversing, Table generation for number 'n', Factorial of a number, Prime number, Factors of a number, Perfect number, Palindrome number, Armstrong number, GCD and LCM of 2 numbers.	6
2	Algorithm 2.1 Concept: Problem Solving. 2.2 Steps in problem solving (Define Problem, Analyze Problem, Explore Solution) Algorithms (Definitions) 2.3 Characteristics of an algorithm 2.4 Time complexity: Big-Oh notation, Omega notation, Theta notation, Efficiency	3
3	Divide and Conquer Method 3.1 General Method, control abstraction. 3.2 Binary search, Breadth First Search, Depth- First Search 3.3 Merge sort, Quick sort	2
4	Greedy Method 4.1 Control Abstraction. 4.2 Knapsack Problem 4.3 Job Sequencing with deadline. Minimum cost spanning trees, Kruskal algorithm, Prim's Algorithm.	4
5	Practical Hands-on 1. DOS Commands 2. MS office Assignment 3. Scratch programming	15
	Total	30

References:

Sr.No.	Title Of the Book	Author's	Publication
1	How to solve it by Computer	R.G.Dromy	Person
2	Fundamentals of Data Structures.	Horowitz and Sahani	Universities Press
3	Introduction to algorithms.	Comen, Lesierson Rivest ,Stein	MIT Press
4	Fundamentals of Computer Algorithms	Sartej Sahani	Universities Press
5	Website:WWW.W3Schools.com	Website	-

SEM II

Subject Code: - 24BBA12101

Subject Name -: Advance Python Programming(2T)

Total Hours :- 30

Total Credits: 02

Prerequisites:

1. Experience with a high-level language (C/C++, Java, MATLAB) is suggested.
2. Prior knowledge of a scripting language (Perl, UNIX/Linux shells) and Object- Oriented concepts is helpful but not mandatory.

Course Objectives:

1. To learn and understand Python programming basics and paradigm.
2. To learn and understand python looping, control statements and string manipulations.
3. Students should be made familiar with the concepts of GUI controls and designing GUI applications.
4. To learn and know the concepts of file handling, exception handling.

Course Outcomes: On completion of the course, student will be able

1. Define and demonstrate the use of built-in data structures “lists” and “dictionary”.
2. Design and implement a program to solve a real world problem.
3. Design and implement GUI application and how to handle exceptions and files.
- 4.

Unit	Details	Lectures
I	Unit 1 : Inheritance 1.1 Single Inheritance 1.2 Multilevel Inheritance 1.3 Multiple Inheritance 1.4 Hybrid Inheritance 1.5 Hierarchical Inheritance IS-A Relationship and HAS-A Relationship	5
II	Unit 2 : Exception Handling and File Handling 2.1 Exception Handling 2.1.1 Python Exception 2.1.2 Common Exception 2.1.3 Exception handling in Python (try-except-else) 2.1.4 The except statement with no exception 2.1.5 Multiple Exception 2.1.6 The try-finally clause 2.1.7 Custom Exception and assert statement 2.2 File Handling 2.2.1 File handling Modes 2.2.2 Writing & Appending to Files 2.2.3 Reading Files Handling File Exceptions	7
III	Unit 3 : GUI Programming with Tkinter 3.1 Introduction 3.2 Tkinter programming 3.3 Tkinter widgets 3.4 Frame 3.5 Button 3.6 Label Entry	8
IV	Unit 4 : Python Libraries 4.1 Statistical Analysis- NumPy, SciPy, Pandas, StatsModels 4.2 Data Visualization- Matplotlib, Seaborn, Plotly 4.3 Data Modelling and Machine Learning- Scikit-learn, XGBoost, Eli5	10

	4.4 Deep Learning- TensorFlow, Pytorch, Keras 4.5 Natural Language Processing (NLP)- NLTK, SpaCy, Python SQL Database Access and Libraries 4.6 Introduction of Database and Libraries(NumPy,SciPy,Pandas) 4.7 Installation 4.8 DB Connection 4.9 Creating DB Table 4.10 INSERT, READ, UPDATE, DELETE operations COMMIT & ROLLBACK operation	
	Total	30

Reference Books:

1. Mark Lutz, Programming Python, O`Reilly, 4th Edition, 2010
- 2 Learning Python, 4th Edition by Mark Lutz
- 3 Programming Python, 4th Edition by Mark Lutz
- 4 Python Programming:An introduction to computer,John Zelle,3rd Edition.

Subject Code: 24BBA12102

Subject: Computer Laboratory Based on Advance Python Programming (2P)

Total Hours :- 30

Total Credits: 02

Objective :

- 1) Remember basics of programming languages and the web development techniques.
- 2) Understands uses of operators, functions, input/output methods.

Outcomes:

- 3) Students will able to do error free applications giving desire output
- Students will able to develop layout as per need of application

Sr. No.	Assignment Name	Practicals
1	Inheritance	8
2	Exception Handling	6
3	Python GUI Programming using Tkinter	6
4	Working with Functions, Modules and Packages	10
Total (Out of 30)		30
Total (Out of 5)		

Assignment on Advance Python

- 1) Write a python program to demonstrate multilevel inheritance by using Base class name as “Team” which inherits Derived class name as “Dev”.
- 2) Write a python program by considering Baseclass as TeamMember and Derived class as TeamLeader use multiple inheritance concept to demonstrate the code.
- 3) Write a python program to make use of issubclass () or isinstance() functions to check the relationships of two classes and instances.
- 4) Write a python program to inherit (Derived class) “course“ from (base class) “University” Using hybrid inheritance concept.
- 5) Write a python program to show the Hierarchical inheritance of two or more classes named as “Square “ & “ Triangle” inherit from a single Base class as “Area “ .
- 6) Define a class named Shape and its subclass (Square/Circle). The subclass has an init function which takes an argument (length/radius). Both classes have an area and volume 48 function which can print the area and volume of the shape where Shape's area is 0 by default.
- 7) Python Program to Create a Class in which One Method Accepts a String from the User and Another method Prints it. Define a class named Country which has a method called print Nationality. Define subclass named state from Country which has a method called print State . Write a method to print state, country and nationality.
- 8) Write a Python Program to depict multiple inheritance when method is overridden in bothclasses and check the output accordingly.
- 9) Write a Python Program to describe a HAS-A Relationship(Composition).
- 10) Define a custom exception class which takes a string message as attribute.
- 11) Write a function called oops that explicitly raises a IndexError exception when called. Then write another function that calls oops inside a try/except statement to catch the error.
- 12) Change the oops function you just wrote to raise an exception you define yourself, called

- MyError, and pass an extra data item along with the exception. Then, extend the try statement in the catcher function to catch this exception and its data in addition to IndexError, and print the extra data item.
- 13) Define a class Date(Day, Month, Year) with functions to accept and display it. Accept date from user. Throw user defined exception "invalidDateException" if the date is invalid.
 - 14) Write text file named test.txt that contains integers, characters and float numbers. Write a Python program to read the test.txt file. And print appropriate message using exception
 - 15) Write a function called safe (func, *args) that runs any function using apply, catches any exception raised while the function runs, and prints the exception using the exc_type and exc_value attributes in the sys module. Then, use your safe function to run the oops function you wrote in Exercises 3. Put safe in a module file called tools.py, and pass it the oops function interactively. Finally, expand safe to also print a Python stack trace when an error occurs by calling the built-in print_exc() function in the standard traceback module (see the Python library reference manual or other Python books for details)
 - 16) Change the oops function in question 4 from SET A to raise an exception you define yourself, called MyError, and pass an extra data item along with the exception. You may identify your exception with either a string or a class. Then, extend the try statement in the catcher function to catch this exception and its data in addition to IndexError, and print the extra data item. Finally, if you used a string for your exception, go back and change it to be a class instance
 - 17) Write a Python GUI program to import Tkinter package and create a window and set its title.
 - 18) Write a Python GUI program to create two buttons exit and hello using tkinter module.
 - 19) Write a Python GUI program to create a Checkbutton widget using tkinter module.
 - 20) Write a Python GUI program to create three single line text-box to accept a value from the user using tkinter module.
 - 21) Write a Python GUI program to create three radio buttons widgets using tkinter module.
 - 22) Write a Python GUI program to create a Listbox bar widgets using tkinter module.
 - 23) Write Python GUI program to display an alert message when a button is pressed.
 - 24) Write Python GUI program to take input of your date of birth and output your age when a button is pressed.
 - 25) Write Python GUI program which accepts a sentence from the user and alters it when a button is pressed. Every space should be replaced by *, case of all alphabets should be reversed, digits are replaced by ?.
 - 26) Write Python GUI A program to create a digital clock with Tkinter to display the time.
 - 27) Write a Python GUI program to implement simple calculator.

Total Contact Hours: -30

Total Credits: - 2

Subject Code: - 24BBA12103

Subject Name -: Relational Database Management System

Objectives:

1. Enables students to understand relational database concepts and transaction management concepts in database system.
2. Enables student to write PL/SQL programs that use: procedure, function, package, cursor and trigger.
3. An RDBMS includes functions that maintain the security, accuracy, integrity and consistency of the data.

Expected Outcome:

CO1: Understanding of various RDBMS products . CO2: To get knowledge of Front End and Backend.
CO3: Understanding of various programming aspects and Writing of compact code (Small Program writing).

Unit No.	Unit Title	Contents	No. of Hours
1.	Introduction To RDBMS	Difference Between DBMS and RDBMS. Relationship among application programs and RDBMS.	2
2.	PL-SQL	Overview of PLSQL PLSQL Block Exception Handling Functions, Procedures Cursor Trigger Package	15
3	Transaction Management	Transaction Concept Transaction Properties Transaction States TCL Commands Concurrent Execution Serializability View Serializability and conflict serializability Problem solving on Transactions	6
4	Concurrency Control & Recovery System	Lock Based Protocol 2PL Protocol Timestamp Based Protocol Deadlock Handling Deadlock Problem Examples Failure Classification Recovery & Atomicity Recovery with concurrent transaction	7
		Total	30

Suggested References:

Sr. No.	Title of the Book	Author/s	Publication	Place
1	Database Management System	Bipin Desai	Galgotia Publications	New Delhi
2	SQL/PLSQL the programming language of oracle	Ivan Bayross	BPB Publications	New Delhi
3	An Introduction to Database Systems Eighth Edition	C. J.Date, A.Kannan, S.Swamynathan	Pearson Publications	North America
4	Database System Concepts 5th Edition	Silberschatz , Korth, Sudershan	McGraw-Hill	New York
5	Beginning of Relational Data model second edition	Sharon Allen	Apress	New York

Total Contact Hours: -30

Total Credits: - 2

Subject Code: - 24BBA12104

Subject Name -: Computer lab based on Relational Database Management System(2P)

Objective :

- 4) Remember basics of programming languages and the web development techniques.
- 5) Understands uses of operators, functions, input/output methods.

Outcomes:

- 6) Students will able to do error free applications giving desire output
- 7) Students will able to develop layout as per need of application.

Topics for Laboratory Assignments For RDBMS	
Sr. No	Assignment
1	Data Type, PLSQL Block and Control Structure
2	Error and Exception Handling
3	Function
4	Procedure
5	Cursors
6	Triggers
7	Package

Sr.No	Assignment
1	<ol style="list-style-type: none">1. Write a PL/SQL block to accept a number and display multiplication table of the given number.2. Write a PL/SQL block which will accept student details, calculate the class using per value and insert the record into Student (rno, sname, class, per, class) table.3. Write a PL/SQL block which will accept two numbers from user, check whether numbers are positive or negative. If positive number then display only the odd numbers between the entered numbers.4. Write a PL/SQL block which will accept roll number of a student and display record of student from student table(use %ROWTYPE attribute)

2	<p>1. Consider the following entities and their relationships. Wholesaler (w_no, w_name, address, city) Product (product_no, product_name, rate) Relation between Wholesaler and Product is Many to Many with quantity as descriptive attribute. Constraint: Primary key, rate should be > 0.</p> <p>2. Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:</p> <ol style="list-style-type: none">1. Write a function to accept quantity from user. Quantity must be within range 50-200. If user enters the quantity out of range then raise an user defined exception “quantity_out_of_range” otherwise enter the record in table.2. Write a PL/SQL block which accept rate from user. If user enters rate less than or equal to zero then raise an user defined exception “Invalid_Rate_Value” otherwise display message “Correct Input”. <p>3. Consider the following entities and their relationships. Student (rollno, sname, class, timetable, mobileno) Lab (LabNo, LabName, capacity, equipment) Relation between Student and Lab is Many to One. Constraint: Primary Key, capacity should not be null.</p> <p>Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:</p> <ol style="list-style-type: none">1) Write a function to accept lab number from user as parameter. ” if user enters invalid lab number then raise an user defined exception “Invalid_Lab_No” otherwise display the student details of the same lab.
---	---

	<p>2) Write a PL/SQL block which accept a mobile number from user. If mobileno less than or more than 10 digits then raise an user defined exception “Invalid_Mobile No” otherwise display the “Correct input...! “.</p>
3	<p>1.Consider the following entities and their relationship. Newspaper (name,language , publisher , cost) Cities (pincode , city, state) Relationship between Newspaper and Cities is many-to-many with descriptive attribute daily required Constraints: name and pincode primary key Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following: 1) Write a procedure to calculate city wise total cost of each newspaper 2) Write a procedure which display details of news papers having cost greater than 2 Rs. 3) Write a procedure which take display details of cities where English news paper is supplied.</p>
4	<p>SET A : Consider the following entities and their relationships. Client (client_no, client_name, address, birthdate) Policy_info (policy_no, desc, maturity_amt, prem_amt, date)Relation between Client and Policy_info is Many to Many Constraint: Primary key, prem_amt and maturity_amt should be > 0 Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following: 1) Write a function which will return total maturity amount of policies of a particular client. 2) Write a function which will return minimum maturity amount of all policies.</p>
5	<p>Consider the following entities and their relationships.Project (pno, pname, start_date, budget, status) Department (dno, dname, HOD, loc) The relationship between Project and Department is Many to One. Constraint: Primary key. Project Status Constraints: C – Completed, P - Progressive, I – Incomplete P Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following: 1) Write a cursor which will display list of projects started in month of</p>

	<p>“January” 2020.</p> <p>2) Write a cursor which will display status wise project details of each department.</p>
6	<p>Consider the following entities and their relationships.</p> <p>Employee (emp_id, emp_name, address)</p> <p>Investment (inv_no, inv_name, inv_date, inv_amount) Relation between Employee and Investment is One to Many.</p> <p>Constraint: Primary key, inv_amount should be > 0.</p> <p>Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:</p> <ol style="list-style-type: none"> 1) Write a trigger which will fire before insert or update on Investment having investment amount less than 10000. (Raise user defined exception and give appropriate message) 2) Write a trigger which will fire before insert or update on Employee having Emp id less than equal to zero (Raise user defined exception and give appropriate message) 3) Write a trigger which will fire before insert or update on Investment having investment date greater than current date. (Raise user defined exception and give appropriate message)
7	<p>Consider the following entities and their relationships.</p> <p>College (code, college_name, address)</p> <p>Teacher (teacher_id, teacher_name, Qualification, specialization, salary, Desg)</p> <p>Relation between Teacher and College is Many to One.</p> <p>Constraint: Primary Key, qualification should not be null.</p> <p>Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:</p> <ol style="list-style-type: none"> 1) Write a package, which consists of one procedure and one function. Pass college code as a parameter to procedure and display details of college. Write a function which will return teacher name having maximum salary. 2) Write a package, which consists of one cursor and one trigger. Cursor should display teachers detail and trigger should fire on Teacher table before insert or update salary having salary less than or equal to zero.

Total Contact Hours: -30

Total Credits: - 2

Subject Code: -24BBA12406

Subject Name -: Web Technologies

Course Objective:

1. To understand the concepts of internet programming.
2. Learn how to develop the overall look of the website.
3. Learn how to use CSS in HTML documents to format web pages.
4. Create error free applications giving desired results.

Course Outcome:

CO1: To understand the basic concept internet programming using HTML.CO2:

To use CSS in HTML code.

CO3: To understand uses of different tags in HTML.

CO4: Apply the syntax and concepts of programming language and generate dynamic web pages

Unit No	Topic	No. of Hours
1	1. Introduction to Protocols 1.1 Clients- Servers and Communication 1.2 Internet-Basic, Internet Protocols (HTTP, FTP, IP) 1.3 World Wide Web(WWW) 1.4 HTTP request message, HTTP response message	5
2	2. Web development basics 2.1 Concepts of effective web design 2.2 Web design issues including Browser and width and Cache 2.3 Display resolution 2.4 Look and Feel of the Website 2.5 Page Layout and linking 2.6 User centric design 2.7 Sitemap 2.8 Planning and publishing website 2.9 Designing effective navigation	8
3	3. HTML 3.1 Introduction to HTML 3.2 Basic HTML Structure 3.3 Common HTML Tags 3.4 Physical and Logical HTML 3.5 Types of Images, client side and server-side Imagemapping 3.6 List, Table, Frames 3.7 Embedding Audio, Video 3.8 HTML form and form elements 3.9 HTML Web Storage 3.10 Introduction to DHTML	12

4	4. Cascading Style Sheets 4.1 Need for CSS 4.2 Introduction to CSS 4.3 Basic syntax and structure 4.4 Using CSS- 4.4.1 background images, colors and properties, 4.4.2 manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS 4.5 Overview and features of CSS2 and CSS3	5
	Total	30

Reference Books:

1. Complete HTML- Thomas Powell
2. HTML and JavaScript – Ivan Bayross
3. HTML & CSS: The Complete Reference, Fifth Edition
4. Web Technologies Black Book- Kogent Learning Solutions Inc. (with Cd)
5. Web Technologies - Godbole, Khate