Fundamentals of Java		Semester	III
Course Code	BBCA301	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	2:0:2	SEE Marks	50
Total Hours of Pedagogy	28 Hours of Theory + 20 Hours of Practical	Total Marks	100
Credits	04	Exam Hours	03
Type of the Corse	Theory + Practical		

Course objectives:

- To learn primitive constructs JAVA programming language.
- To understand Object Oriented Programming Features of JAVA.
- To gain knowledge on: packages, exceptions and multithreaded programing.

Module-1

An Overview of Java: Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOPPrinciples), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords).

Data Types, Variables, and Arrays: The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays.

Operators: Arithmetic Operators, Relational Operators, Boolean Logical Operators, The AssignmentOperator, The ? Operator, Operator Precedence, Using Parentheses.

Control Statements: Java's Selection Statements (if, The Traditional switch), Iteration Statements(while, do-while, for, The For-Each Version of the for Loop, Nested Loops), Jump Statements (Using break, continue, return)

Module-2

Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables,Introducing Methods, Constructors, The this Keyword, Garbage Collection.

Methods and Classes: Overloading Methods, Objects as Parameters, Argument Passing, ReturningObjects, Recursion, Access Control, Understanding static, Introducing final

Module-3

Inheritance: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Local Variable Type Inference and Inheritance, The Object Class.

Interfaces: Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods.

Module-4

Exceptions: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try andcatch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions,Creating Your Own Exception Subclasses, Chained Exceptions

Module-5

Packages: Packages, Packages and Member Access, Importing Packages.

Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, CreatingMultiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, InterthreadCommunication, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State.

Teaching Methodology: Chalk and talk method / PowerPoint Presentation.

Course outcome (Course Skill Set)

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

CO 1. Demonstrate proficiency in writing simple programs involving branching and looping structures.
CO 2. Design a class involving data members and methods for the given scenario.
CO 3. Apply the concepts of inheritance and interfaces in solving real world problems.
CO 4. Use the concept exception handling in solving complex problem
CO 5. Apply concepts of packages and multithreading in program development

Suggested Learning Resources:

Textbooks:

1. Java: The Complete Reference, Twelfth Edition, by Herbert Schildt, November 2021, McGraw-Hill, ISBN: 9781260463422

Reference Books

1. Programming with Java, 6th Edition, by E Balagurusamy, Mar-2019, McGraw Hill Education, ISBN: 9789353162337.
2. Thinking in Java, Fourth Edition, by Bruce Eckel, Prentice Hall, 2006

Web links and Video Lectures (e-Resources):

- https://onlinecourses.nptel.ac.in/noc22_cs47/preview
- https://www.youtube.com/watch?v=3a1FXBR6QXY&list=PLZ2ps__7DhBaXTORd_OKXZRXCMh3dGnBZ&index=1

Practical content for IPCC

1. Write a Java program that uses a for loop to calculate the sum of even numbers between 1 and 100. The program should use continue to skip odd numbers and print the final sum of even numbers after the loop finishes.
2. Write a Java program to input a number and determine if it is prime or not.
3. Write a Java program to create a class Student with members name and rollNumber. Define a constructor to initialize these members. Then, create a method displayDetails() to display the student's details.
4. Write a Java program to define a class Book with attributes title, author, and price. Use a constructor to initialize these attributes. Demonstrate the use of the this keyword to refer to instance variables and display the book details.
5. Write a Java program to demonstrate the use of the static keyword by creating a Counter class with a static variable count that tracks the number of instances of the class. Also, define a final variable to hold a constant value such as the maximum allowed instances.
6. Write a Java program that demonstrates method overriding and dynamic method dispatch. Create a base class Shape with a method area(). Override the area() method in derived classes Circle and Rectangle. Use dynamic method dispatch to call the area() method from the base class reference.
7. Write a Java program to create an interface Shape with methods area() and perimeter(). Implement this interface in two classes: Circle and Rectangle. In each class, implement the methods to calculate and display the area and perimeter. Instantiate the classes and display the results for both shapes.
8. Write a Java program that accepts an array of integers and attempts to access the elements using an invalid index. Use multiple catch clauses to handle ArrayIndexOutOfBoundsException and NumberFormatException (if the user inputs a non-integer value).
9. Write a Java program that creates a package named shape. In this package, define a classe: Rectangle. The class should have methods to calculate the area and perimeter. In the main class, import the shape package and access the methods to calculate area and perimeter of shape.
10. Write a Java program to create two threads. The first thread should print numbers from 1 to 5, and the second thread should print the alphabet letters from A to E. Start both threads simultaneously and ensure that both print their respective outputs.

Introduction to Python		Semester	III
Course Code	BBCA302	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 Hrs	Total Marks	100
Credits	03	Exam Hours	03

Course objectives:

- Understand basics of Python programming
- Implement various data structures in Python programming
- Understand implementation of OOP concepts using Python Programming.

Module-1

Introduction: Python programming, history, features, Installing Python, Python Interpreter, IDLE, Keywords, Variables, Constants, Literals, Data Types, Data type conversion, Indentation, Comments, expression, statements, Input & Output functions, range() function , Operators: assignment (single value, multiple values), arithmetic, relational, logical.

Module-2

Conditionalstatements: if, if-else, elif, nested if; **Looping statements**: while, for, for each, nested loops; break, continue; Else statement with loop; **Functions Basics**: Built-in Functions, Declaring and calling user defined functions, Parameters and default arguments, Fruitful functions and void functions, recursive functions.

Module-3

Strings: Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings-Concatenation, Comparison, Slicing and Joining, Traversing; Python built-in String Methods.
List: Creating Lists; Operations on Lists; Built-in Functions on Lists, Python built-in Methods, Nested Lists

Module-4

Tuple: Creating Tuples; Operations on Tuples; Built-in Functions on Tuple;
Set: Creating Sets; Operations on Sets; Built-in Functions on Sets; Set Methods.
Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods;

Module-5

Object Oriented Programming: Defining a class, Instance members, Constructors, Inheritance (single, multi-level, multiple), Special methods (`__init__(), __str__()`), Polymorphism (method overloading, operator overloading)

Teaching Methodology: Chalk and talk method / PowerPoint Presentation.

Course outcome (Course Skill Set)

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

1. Understand basics of Python programming
2. Understand conditional, looping and function concepts
3. Apply string and list data structures to solve real world problems.
4. Apply tuple, set, and dictionary data structures to solve real world problems.
5. Understand implementation of OOP concepts using Python Programming.

Suggested Learning Resources:

Books

1. Zero to Mastery In Python Programming by Rakesh K. Yadav, Srinivas Arukonda, Monu Singh, 2nd Edition, Vayu Education of India
2. Think Python How to Think Like a Computer Scientist, Allen Downey et al., 2 nd Edition, Green Tea Press. Freely available online @https://www.greenteapress.com/thinkpython/thinkCSpy.pdf, 2015.
3. Introduction to Python Programming, Gowrishankar S et al., CRC Press, 2019.

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=8ndsDXohLMQ&list=PLZ2ps__7DhBb2cXAu5PevO_mzgS3Fj3Fs

Database Management System		Semester	III
Course Code	BBCA303	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 Hrs	Total Marks	100
Credits	03	Exam Hours	03
Type of the Corse	Theory		

Course objectives:

- To Provide a strong foundation in database concepts, technology, and practice.
- To Practice SQL programming through a variety of database problems.
- To Understand the relational database design principles.
- To Demonstrate the use of concurrency and transactions in database.
- To Design and build database applications for real world problems.
- To become familiar with database storage structures and access techniques.

Module-1

Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using theDBMS approach, History of database applications.

Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schemaarchitecture and data independence, database languages, and interfaces, The Database System environment.**Conceptual Data Modelling using Entities and Relationships**: Entity types, Entity sets, Attributes, Keys, st Relationship Types, Relationship Sets, Roles, ructuralconstraints, Weak entity types, ER diagrams,Specialization and Generalization.

Module-2

Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas,Update operations, transactions, and dealing with constraint violations.

Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relationalmapping.

Relational Algebra: Unary and Binary relational operations, additional relational operations (aggregate,grouping, etc.) Examples of Queries in relational algebra.

Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relationalmapping.

Module-3

Normalization: Database Design Theory – Introduction to Normalization using Functional and MultivaluedDependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms basedon Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency andFourth Normal Form, Join Dependencies and Fifth Normal Form.

Module-4

SQL: SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL,retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL

Advanced SQL: More complex SQL retrieval queries, Specifying constraints as assertions andaction triggers, Views in SQL.

Module-5

Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirableproperties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules basedon Serializability, Transaction support in SQL.

Concurrency Control in Databases: Two-phase locking techniques for Concurrency control, Concurrencycontrol based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrencycontrol techniques, Granularity of Data items and Multiple Granularity Locking.

Teaching Methodology: Chalk and talk method / PowerPoint Presentation.

Course outcome (Course Skill Set)

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

CO 1. Describe the basic elements of a relational database management system

CO 2. Design entity relationship for the given scenario.

CO 3. Analyse various normalization forms for the given application.

CO 4. Apply various Structured Query Language (SQL) statements for database manipulation.

CO 5. Understandd transaction processing and concurrency control in database.

Suggested Learning Resources:

Textbooks:

1. Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=OMHbGm9SQuE&list=PLZ2ps__7DhBYc4jkUk_yQAjYEVFzVzhdU

Computer Networks		Semester	III
Course Code	BBCA304	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 Hrs	Total Marks	100
Credits	03	Exam Hours	03
Type of the Corse	Theory		

Course objectives:

- Study the TCP/IP protocol suite, switching criteria and Medium Access Control protocols for reliableand noisy channels.
- Learn network layer services and IP versions.
- Discuss transport layer services and understand UDP and TCP protocols.
- Demonstrate the working of different concepts of networking layers and protocols.

Module-1

Introduction: Data Communications, Networks, Network Types.
Networks Models: ProtocolLayering, TCP/IP Protocol suite, The OSI model,
Transmission Media: Transmissionmedia, Guided Media, Unguided Media.

Module-2

Data Link Layer: **Error Detection and Correction**: Introduction, Block Coding, Cyclic Codes.
Datalink control: DLC Services: Framing, Flow Control, Error Control, Connectionless and ConnectionOriented, Data link layer protocols, High Level Data Link Control.
Media Access Control: Random Access, Controlled Access

Module-3

Network Layer: Network layer Services, Packet Switching, IPv4 Address, IPv4 Datagram, IPv6 Datagram.
Routing Algorithms: Unicast Routing Protocols: DVR, LSR, PVR,Unicast Routing protocols: RIP, OSPF, BGP, Multicasting Routing-MOSPF.

Module-4

Transport Layer: Introduction, Transport-Layer Protocols: Introduction, UserDatagram Protocol, Transmission Control Protocol: services, features, segments, TCP connections,flow control, Error control, Congestion control.

Module-5

Application Layer: Introduction, Client-Server Programming
Standard ClientServer Protocols: World Wide Web and HTTP, FTP, Electronic Mail, Domain Name System(DNS), TELNET, Secure Shell (SSH)

Teaching Methodology: Chalk and talk method / PowerPoint Presentation.

Course outcome (Course Skill Set)
At the end of the course, the student will be able to :

CO 1. Explain fundamentals of computer networks.
CO 2. Understand error ditection and control in data link layer.
CO 3. Apply routing protocols in network layer.
CO 4. Analyze the principles of protocol layering in modern communication systems.
CO 5. Demostrate various client-server protocols in application layer.

Suggested Learning Resources:

Books

1. Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, Tata McGraw-Hill,2013.

Reference Books:

1. Larry L. Peterson and Bruce S. Davie: Computer Networks – A Systems Approach, 4th Edition, Elsevier, 2019.
2. Nader F. Mir: Computer and Communication Networks, 2nd Edition, Pearson Education, 2015.
3. William Stallings, Data and Computer Communication 10th Edition, Pearson Education, Inc.,2014.

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=3DZLItfbqtQ&list=PL06E2D9AEF82AE966

Operating System Concepts		Semester	III
Course Code	BBCA305	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 Hrs	Total Marks	100
Credits	03	Exam Hours	03
Type of the Corse	Theory		

Course objectives:

- To Demonstrate the need for OS and different types of OS
- To discuss suitable techniques for management of different resources
- To demonstrate different APIs/Commands related to processor,memory, storage and file system management.

Module-1

Introduction to operating systems: What operating systems do; Computer Systemorganization; Computer System architecture; Operating System structure; Operating System operations;Process management; Memory management; Storage management; Protection and Security; Distributedsystem; Special-purpose systems; Computing environments.

System structures: Operating System Services: User - Operating System interface; System calls; Types of system calls;System programs; Operating system design and implementation; Operating System structure; Virtualmachines; Operating System debugging, Operating System generation; System boot.

Module-2

Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication

Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues.

Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling; Multiple-processor scheduling

Module-3

Process Synchronization: Synchronization: The critical section problem; Peterson's solution;Synchronization hardware; Semaphores; Classical problems of synchronization;

Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Module-4

Memory ManagementStrategies: Background; Swapping; Contiguous memoryallocation; Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

Module-5

File System: File concept; Access methods; Directory and Disk structure; File system mounting; File sharing;

System Protection: Goals of protection, Principles of protection, Domain ofprotection, Access matrix.

System Security: The Security Problem, Program Threats, System and Network Threats

Teaching Methodology: Chalk and talk method / PowerPoint Presentation.

Course outcome (Course Skill Set)

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

CO 1. Understand opaerating system and its structure.

CO 2. Under process management and apply process scheduling algorithms

CO 3. Analyse the various techniques for process synchronization and deadlock handling.

CO 4. Apply the various techniques for memory management.

CO 5. Explain file system concept and describe the need for system protection & security.

Suggested Learning Resources:

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Reference Books

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE),2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=jciGIvn7UfM&list=PLyqSpQzTE6M9SYI5RqwFYtFYab94gJpWk

Python Lab		Semester	III
Course Code	BBCAL306	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	1:0:2	SEE Marks	50
Total Hours of Pedagogy	14 sessions	Total Marks	100
Credits	02	Exam Hours	03
Type of the Course	Practical		

Course objectives:

- To understand basics of Python programming
- To implement various data structures in Python programming
- To understand implementation of OOP concepts using Python Programming.

1. Write a Python program to input two numbers from the user, perform basic arithmetic operations (addition, subtraction, multiplication, division), and display the results.
2. Write a Python program to implement bank operations such as deposit, withdrawal, and balance using conditional statement.
3. Write a Python program to implement password authentication using looping statement.
4. Write a Python program that defines a void function print_table() which prints the multiplication table of a number up to N.
5. Write a Python program to print length of a string without using built-in function.
6. Write a Python program that accepts a string of numbers separated by spaces from the user. Convert the string into a list of integers and calculate the sum of the numbers. Use the range() function to generate a sequence of numbers and calculate the product of the first five numbers.
7. Write a Python program that creates a list of 5 integers. Perform the following operations: append a new element, remove an element, and display the list after each operation.
8. Write a Python program that demonstrates the use of nested tuples. Create a tuple that contains another tuple inside it. Access elements from both the outer and inner tuples using indexing. Then, perform tuple unpacking on a nested tuple to assign values to variables.
9. Write a Python program that performs the following operations on sets: union, intersection, difference, and symmetric difference on two sets. Display the results for each operation.
10. Write a Python program to create a dictionary. Perform the following operations on a dictionary: retrieve a value using a key, remove a key-value pair, and check if a key exists in the dictionary.
11. Write a Python program to implement single inheritance. Create a base class Person with attributes name and age. Then, create a derived class Student that inherits from Person and adds an additional attribute student_id. Display the details of a student.
12. Write a Python program to demonstrate operator overloading. Define a class Rectangle with methods to calculate area. Overload the + operator to add the areas of two rectangles

Course outcome (Course Skill Set)

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

CO 1. Understand and implement basic Python syntax, input/output operations, and arithmetic operators.
CO 2. Apply control flow structures like conditionals and loops to implement real-world applications.
CO 3. Work with data structures in Python, including lists and strings
CO 4. Work with data structures in Python, including tuples, sets, and dictionaries.
CO 5. Understand object-oriented programming concepts such as inheritance and operator overloading.

Database Management System Lab		Semester	III
Course Code	BBCAL307	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	1:0:2	SEE Marks	50
Total Hours of Pedagogy	14 sessions	Total Marks	100
Credits	02	Exam Hours	03
Type of the Course	Practical		

Course objectives:

- To Provide a strong foundation in database concepts, technology, and practice.
- To Practice SQL programming through a variety of database problems.
- To Understand the relational database design principles.
- To Demonstrate the use of concurrency and transactions in database.
- To Design and build database applications for real world problems.
- To become familiar with database storage structures and access techniques.

Sl. No.	Experiment
1	Create a table called Employee that contain attributes EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION& execute the following: 1. Create a user and grant all permissions to theuser. 2. Insert the any three records in the employee table contains attributes EMPNO,ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback.Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert null values to the employee table and verify the result.
2	Create a table called Employee that contain attributes EMPNO,ENAME,JOB, MGR,SAL & execute the following: 1. Add a column commission with domain to the Employeetable. 2. Insert any five records into the table. 3. Update the column details of job 4. Rename the column of Employ table using alter command. 5. Delete the employee with specific Empno.
3	Execute the queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by,Orderby. Employee(E_id, E_name, Age, Salary) 1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employeetable 3. Find the Maximum age from employee table. 4. Find the Minimum age from employeetable. 5. Find salaries of employee in Ascending Order. 6. Find grouped salaries of employees.
4	Create a row level trigger for the customers table that would fire for INSERT or UPDATE orDELETE operations performed on the CUSTOMERS table. This trigger will display thesalary difference between the old & new Salary. CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)
5	Create cursor for Employee table & extract the values from the table. Declare the variables,Open the cursor &extrct the values from the cursor. Close the cursor. Employee(E_id, E_name, Age, Salary)

Course outcome (Course Skill Set)

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

CO 1. Describe the basic elements of a relational database management system

CO 2. Design entity relationship for the given scenario.

CO 3. Analyse various normalization forms for the given application.

CO 4. Apply various Structured Query Language (SQL) statements for database manipulation.

CO 5. Understandd transaction processing and concurrency control in database.