SEMESTER-VI

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – VI of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF COMPUTER SCIENCE

Major Course – 1: Basic and Advanced Java (Theory)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-Requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Core java with JDBC (Theory)	4	0	0	Students with basic coding knowledge	Nil

Course Outcome:

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

- Clarify the concept of Object-Oriented Programming and describe the basic concepts of Java Programming Language, creation of the class and its objects.
- Describe the concepts of looping, String, Arrays and Wrapper classes.
- Clarify the concepts of Exception handling techniques, Inheritance and Interface.
- Describe the concepts of Package, Multithreading and JDBC

UNIT 1 Java Introduction :

- Creating first java classes
- Brief Introduction to Object Oriented Programming Concept
- Learning about Java
- Features of Java
- Analyzing a java application that uses console output
- Adding comments to a java
- Saving, compiling and running a java application
- Creating a java application using GUI output and GUI input

Naming convention for Constants, variables and Identifiers

Data Types

- Integer
- Floating point
- Character
- Boolean

Understanding numeric type conversion

Operators in Java

- Arithmetic
- Relational (Comparison operators)
- Boolean Logical
- Increment and Decrement
- Conditional
- Bitwise
- Understanding precedence

Using methods, classes and objects

- Creating methods with arguments
- Class concepts and creating a class
- Creating instance methods in a class
- Declaring objects and using their methods
- Static method and Static variable
- Understanding block and scope
- Method overloading
- Constructors
- Sending arguments to constructors
- Constructors overloading
- 'this' keyword
- Working with constants

UNIT 2 **Decision Making, Looping, Strings, Arrays,Wrapper Classes and Inheritance** Stack

Brief overview about

- if and if…..else, Nesting if… else
- switch statement
- while loop, for loop, do…while loop

Characters, String class and String Buffered class

- Manipulating characters class
- isUpprCase(), toUpperCase(), isLowerCase(), toLowerCase()
- isDigit(), isLetter(), isLetterOrDigit(), isWhitespace()
- Manipulating String class
- Declaring a String Object
- Comparing String values
- toUpperCase() , toLowerCase()
- length(), indexOf(), charAt(), endswith(), startWith()
- replace(), toString()
- Manipulating StringBuffer class
- setLength(), capacity(), append(), insert()
- setChartAt(), charAt()

Arrays

- Declaring and initializing an array
- Using subscripts with an array
- Passing array to methods
- Creating arrays of strings

Inheritance

- Concept of inheritance
- Extending classes

- Method overriding
- Constructor calling during inheritance
- Super class constructor that requires arguments (using 'super' keyword)
- Accessing super class methods (using 'super' keyword)
- Method which cannot be override
  - 'final' method
  - 'final' super class
  - Static method

UNIT 3 **Exception Handling, Inheritance and Packages**

Interfaces and Abstract Classes

- Defining Abstract class
- Using Abstract class
- Defining Interfaces
- Implementing Interfaces
- Multiple inheritance using Interfaces

Wrapper Classes (Overview**)**

- Byte class, short class, Integer class, Long class, Float class, Double class, Boolean class

Packages

- Define a Package
- Creating a Package
- Class and package
- Import statement
- Importing a Package
- Brief introduction of Access Protection (Access modifiers)

Excepting Handing

- Learning about exceptions
- Trying code and catching exceptions
- Throwing and catching multiple exceptions
- 'finally' block
- Checked and unchecked exception(overview only)
- Creating own exceptions (custom exception)

UNIT 4 **Multithreading and JDBC**

Multithreading

- Introduction
- Thread Life Cycle
- Creating and running thread (using Thread class and Runnable interface)
- Thread Priorities
- Thread join(), sleep() method

Introduction to JDBC

- What is JDBC?
  - A quick overview of JDBC and its role in Java

applications.

- JDBC Architecture and drivers
  - Basic understanding of the JDBC architecture and introduction to the types of JDBC drivers
- Connecting to the Database
  - DriverManager and Connection
    Understanding how to use `DriverManager.getConnection()` to establish a connection.
- Executing SQL Queries
  - Statement Interface
    Introduction to Statement for executing SQL queries.
  - Simple Queries
    Running basic SQL commands (SELECT, INSERT, UPDATE, DELETE).
- Working with Result Sets
  - ResultSet Interface
    Learn how to retrieve and navigate data from a ResultSet using methods like next(), getString(), and getInt().
  - Handling Data
    Working with multiple rows and different data types.
- Prepared Statement
  - Introduction to Prepared Statements
  - Difference between Statement and Prepared Statement and how to use placeholders

Textbook:

JAVA for Beginners
Publication : Cengage
Learning By: Joyce
Farrell

Reference Book:

1. Object Oriented
   Programming in java
   Publication : Dreamtech
   By Dr. G.T.Thampi

2. JAVA
   Programmin
   g
   Publication:
   Pearson

By Hari Mohan Pandey
3. Client/Server Programming with JAVA and CORBA Second edition

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – VI of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF COMPUTER SCIENCE

Major Course – 2: Database Management System -II

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-Requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Database Management System - II	2	0	2	10 + 2 from a recognized board in any stream	Nil

Course Outcome:

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

- Recognize the elements of Database for real life applications.
- Familiar with the advanced database concepts such as distributed database, business intelligence and data warehouse etc. COBC4503.04: Describe the concepts of
- Get familiar with the fourth generation language named structure query language which can be used to solve ad hoc queries.
- Experience to design database table and establish relationship between them.

UNIT 1 **Introduction to SQL**

Data Definition Commands

- Data Types
- Creating Table Structures
- SQL Constraints
  - Primary Key
  - Foreign Key
  - Not Null
  - Unique
  - Default
  - Check

Data Manipulation Commands

- Adding Table Rows
- Saving Table Changes

- Listing Table Rows
- Updating Table Rows
- Restoring Table Contents
- Deleting Table Row

Select Query

- With Conditional Restrictions
- Arithmetic Operators
- Logical Operators
- Special Operators

Advanced Data Definition Commands

- Changing a Column's Data Type
- Changing a Column's Data Characteristic
- Adding a column
- Dropping a column
- Advanced Data Update
- Copying Parts of Table
- Adding Primary and Foreign Key Designations
- Deleting Table from The Database

UNIT 2 **Business Intelligence and Data Warehouse**

- The need for data analysis
- Business Intelligence
- Business Intelligence Architecture
- Decision Support Data
- Operational Data Vs. Decision Support Data
- Decision Support Database Requirements
- The Data Warehouse
- Online Analytical Processing
- Multidimensional Data Analysis Techniques
- Advanced Database Support o Easy-To-Use End-User Interface
- Client/Server Architecture
- Data Mining

UNIT 3 **Distributed Database Management System**

Introduction of (DDBMS)Distributed Database Management Systems

- Evolution of DDBMS
- Distributed Processing and Distributed Database
- DDBMS Advantages and Disadvantages
- Characteristics of DDBMS
- Components of DDBMS

Levels of Data and Process Distribution

- Single-Site Processing, Single-Site Data (SPSD)
- Multiple-Site Processing, Single-Site Data (MPSD)
- Multiple-Site Processing, Multiple-Site Data (MPSD)

Distributed Database Transparency Features

Distributed Transparency

Transaction Transparency

- Distributed Requests and Distributed Transactions
- Distributed Concurrency Control

- Two-Phase Commit Protocol
- Performance Transparency and Query Optimization

UNIT 4 **Advance SQL**

Aggregate Functions

- Min
- Max
- Count
- Sum
- Avg

Set Operators

- Union
- Union All
- Intersect
- Minus

SQL Join

- Cross Join
- Natural Join
- Join Using Clause
- Join On Clause
- Outer Join

SQL Functions (Demonstrate the use of SQL functions using SQL query on different tables.)

- Date and Time
- Numeric
- String
- Conversion

Subqueries (Demonstrate the use of sub queries on different tables.)

- Where Subqueries
- In Sub queries
- Multirow Subquery Operators: Any and All
- From Subqueries
- Attribute list Subqueries
- Correlated Subqueries

Sequence (Create sequences and demonstrate the use of sequence.)

View

Text Book:

Database System Concepts (First Edition: 2008) Publisher: Cengage Learning By Peter Rob and Carlos Coronel

Chapter-12 (12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.9, and 12.10)), Chapter-13 (13.1, 13.2, 13.3, 13.4,(13.4.1, 13.4.2), 13.5, 13.6(13.6.1, 13.6.2, 13.6.3, 13.6.4), 13.9) Excluding (13.5.1, 13.5.2, 13.6.5, 13.6.6, 13.6.7, 13.6.8,13.7, 13.8, 13.10) Chapter-7 (7.1, 7.2 (7.2.4, 7.2.5, 7.2.6, 7.2.7) 7.3, 7.4, 7.5, 7.6.3) Excluding (7.1.1, 7.1.2, 7.2.3) Chapter-8 (8.1, 8.2, 8.3, 8.4, 8.5)

Reference Books:

1. Introduction to Database Management Systems (First Edition 2006) Publisher: Tata McGraw-Hill By ISRD Group

2. An Introduction to Database Systems (Eighth Edition 2006) Publisher : Pearson By C. J. Date, A. Kannan & S. Swamynathan

3. An Introduction to Database Systems Publisher: Pearson By ITL Education Solutions Limited.

St. Xavier's College (Autonomous), Ahmedabad

FACULTY OF COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

Major Course – 3: Core and Advanced Java (Practical)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-Requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Core java with JDBC (Practical)	0	0	4	Students with basic knowledge of coding	Nil

Course Outcome:

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

- Apply the various concepts of java in the development of basic java programs with or without use of the objects
- Develop programs by using the concepts of java loop constructs, String, Array and inheritance.
- Create the programs by using the concepts of Exception handling, Wrapper classes, abstract class, Interface and package.
- Apply the concepts of multithreading in the program development and also demonstrate the use of JDBC.

UNIT - 1

- Write a program to calculate the hypotenuse of right angled triangle when other sides of the triangle are given. (Hypotenuse = square root (x*x + Y *Y))
- Create a complex number class. The class should have a constructor and methods to add, subtract and multiply two complex numbers and to return the real and imaginary parts.
- Write a program using the arithmetic operators to perform algebraic operations on two numbers. (Algebraic operation is +, - , *, /, %)
- Write a program to calculate the area of rectangle and square by overloading the area method. The area method should be static. The values required for rectangle and square should be accepted from the user.
- Write a java program to create a class Circle. It should have three int members x,y(center co-ordinates) and radius. Program should contain three constructors. First no argument constructor should initialize center at origin with radius 1. Second single argument constructor accepts the value of radius and should be centered at origin. Third constructor should accept three arguments and initialize

all three members with them respectively. Also write an appropriate main method to demonstrate the use of each constructor.

- Write a program to evaluate simple interest of a given principle, rate and time.[I=P*R*T]
- Write a java program to display powers of 2 i.e. 2,4,8,16 etc. up to 1024 using bitwise operators.
- Write a program to convert centimeters to inches.
- The employee list for a company contains employee code, name, designation and basic pay. The employee is given a house rent allowance (HRA) of 10% of the basic pay and dearness allowance (DA) of 45% of basic pay. The total pay of the employee is calculated as basic pay + HRA + DA. Write a class to define the details of the employee. Write a constructor to assign the required initial values. Add a method to calculate HRA, DA and total pay and print them out. Write another class with a main method. Create objects for three different employees and calculate the HRA, DA and total pay.
- Write a java program to scan 3 integer values from the user and display the minimum using conditional operator.

UNIT -2

- Write a java program which will accept one string from the user and display individual vowel count of each vowel as well as total vowels.
- Write a java program which will accept 10 BCA college names from the user and sort them using compareTo() method.
- Write a java program that accepts a string from users and display each character on separate line in reverse order.
- Write a java program which will display the Floyd triangle.
- Write a program to sort a list of students on the basis of the marks.
- Write a program to create a character array to store 6 characters. Also initialize the array with 6 random characters. Now create another array containing 10 characters. Copy the elements ranging from index 2 to 4of first array to second array at the same index.
- Write a program to print prime numbers between 1to100.
- Write Java program to check whether a given string is palindrome or not and display appropriate message.
- Write a program to find sum and multiplication of two matrices A(3,3) and B(3,3).
- Create a class called NumberData that accept any array of the five numbers. Create a sub class called Numplay which provides methods for followings:
  1. Display numbers entered.
  2. Sum of the number.
  3. Average of the numbers.
  4. Maximum of the numbers.
  5. Minimum of the numbers.

  Create a class that provides menu for above methods. Give choice from the command-line argument.
- Write a java program which demonstrates the concept of method overriding. An abstract class Shape contains the method area() which returns an int value.

Square and Rectangle both are the subclasses of the abstract class Shape. Square and Rectangle classes constructor contain one and two arguments respectively for initializing their instance variables. Both the classes will override the area() method declared in the super class Shape. Program should calculate the area of the Square and Rectangle class and it should also display the area along with the value of the instance variables of the Square and Rectangle class.

UNIT - 3

- Write a java program which takes 2 arguments - a string and its length. If the length of the string is not according to given one then throw the user defined LengthMatchException and handles it appropriately.
- Write a java program that accepts 5 positive numbers from user interactively. If any of the number is negative then throw custom exception NegativeException and handle it.
- A method named add() accepts an array of strings as its argument. It converts these to double values and returns their sum. The method generates a NumberFormatException if an element is incorrectly formatted. It can also create and throw a custom exception, RangeException, if an element is less than 10 or greater than 100. Write a program that illustrates how to declare and use this method. Invoke the method from main(). Catch any exceptions that are thrown and display an informative message for the user. Also, provide a finally clause to thank the user for using the program.
- Write a program that demonstrates the instanceof operator. Declare interfaces I1 and I2. Interface I3 extends both of these interfaces. Also declares interface I4. Class X implements I3. Class W extends X and implements I4. Create an object of class W. Use the instanceof operator to test if that object implements each of the interfaces and is of type X.
- Write an interface called Exam with a method Pass(int mark) that returns a Boolean. Write another interface called Classify with a method Division(int average) which returns a string. Write a class called Result which implements both Exam and Classify. The pass method should return true if the marks is greater than or equal to 35 else false. The division method must return “First” when the parameter average is 60 or more, “second” when average is 50 or more but below 60, “no division” when average is less than 50.
- Write a java class called Sumn that finds the sum of all numbers from 1 upto a given number n. This class is contained in the sub directory named Addition , under the current directory where all other java program files are stored. Write a package statement inside the class Sumn and imports it in another file and display the summation of numbers.
- Write a Java program to check whether a string is a valid password, otherwise throw the custom exception, “Invalid Password”. Accept a string using command-line arguments. Password rules are as follows:
    - A password must have at least ten characters.
    - A password consists of only letters and digits.
    - A password must contain at least two digits.

- The abstract class Fish declares one abstract method named display(). It also has two abstract subclasses named FreshWaterFish and SaltWaterFish. Define three concrete subclasses and Trout, Flounder and Tuna. Trout extends FreshWaterFish and Flounder and Tuna both extends SaltWaterFish. Write a main() method which creates an array of type Fish. Instantiate and assign different types of Fish to the elements of the array. Display the name of fish which is of type SaltWaterFish using display() method.
- The class JetPlane declares one abstract method named numEngines(). Therefore, the class itself must also be declared abstract. There are two concrete subclasses named DC8 and DC10. Each of these provides a different implementation of the numEngines() method. The main() method instantiates each of these classes and invokes its numEngines() method.
- Write an application that converts between meters and feet. Its first command line argument is a number and second command line argument is either "centimeter" or "meter". If the argument equals "centimeter" displays a string reporting the equivalent number of meters. If this argument equals "meters", display a string reporting the equivalent number of centimeter. If unit is not given properly then generate custom exception Unitformatexception. If first argument is not proper format then generate numberformatexception. Generate other exception as per requirements.(1 meter=100 centimeter)

UNIT - 4

- Write an application that starts 3 threads. Each thread is instantiated from the same class and displays a message. First thread should display the message “CORE JAVA EXAM” every 300 ms. Second thread should display “SY BCA EXAM” every 500 ms. Third thread should display “UNIVERSITY EXAM” every 400 ms. Create the threads using Runnable interface.
- Create two threads, one thread should print odd numbers between x to y (inclusive) and second should print even numbers between them. Values of x and y should be user given. Main thread should wait for thread to be finished and afterwards it should print the message “Threads Completed”. Create threads with the help of Thread class.
- Write a java program 1” at every 1000 Milliseconds and other should display “Thread 2” at every 3000 milliseconds to create 3 threads using Runnable interface. Three threads should calculate the sum of 1 to 5, 6 to 10 and 11 to 15 respectively. After all thread finishes main thread should print the sum and average.
- Write an interface called Numbers, with a method int Process(int x, int y). Write a class called Sum, in which the method Process finds the sum of two numbers and returns an int value. Write another class called Average, in which the Process method finds the average of the two numbers and returns an int.
- Write a JDBC program which will connect and insert into a student table.
- Write a JDBC program which will connect and apply CRUD operation on an employee table.
- Write a JDBC program which will connect and apply CRUD operation on an doctor table.
- Write a JDBC program which will display username and his IDNumber from a table.
- Write a JDBC program which will show how to update the record from a table.
- Write a JDBC program which will show the usage of PreparedStatement for Insertion.

- Write a JDBC program which will show the usage of PreparedStatement for selection.
- Write a JDBC program which will show the demo of how to handle multiple rows.

Textbook:

JAVA for Beginners
Publication : Cengage
Learning By: Joyce
Farrell

Reference Book:

1. Object Oriented
   Programming in java
   Publication : Dreamtech
   By Dr. G.T.Thampi

2. JAVA Programming
   Publication: Pearson
   By Hari Mohan Pandey

3. Client/Server Programming with JAVA and CORBA Second edition
   Publisher: WILEY
   By Robert Orfali and Dan Harkey