CMR Institute of Technology, Bengaluru		
Department(s): Department of ISE		
Semester: 06	Section(s): A, B, C	Lectures/week: 04
Subject: Advanced Java		Code: BIS402
Course Instructor(s): Dr. S. Seetha, Prof. Varsha		
Course duration: 23 Feb 2026 to 31 May 2026		

Advanced Java Workbook

SUBJECT CODE: BIS402

Dept. of ISE

CMR Institute of Technology, Bengaluru

Table of Contents

Module 1: Collections Framework

MCQs

1. **Which interface in the Collections Framework does not allow duplicates?**
   A: List
   B: Set
   C: Map
   D: Queue

   Ans:

   What does the Iterator interface provide?
   A: Random access to elements
   B: A method to traverse collections
   C: Sorting of elements
   D: Synchronization of collection access

   Ans:

2. **Which class implements a resizable array?**
   A: ArrayList
   B: LinkedList
   C: HashSet
   D: TreeMap

   Ans:

3. **What is the primary benefit of using a LinkedList over an ArrayList?**
   A: Better random access
   B: Faster insertion and deletion
   C: Sorted order by default
   D: Thread safety

   Ans:

4. **Which collection class is synchronized by default?**
   A: ArrayList
   B: Vector
   C: HashSet
   D: LinkedList

   Ans:

5. **Which method removes the current element during iteration?**
   A: list.remove()
   B: Iterator.remove()
   C: Collection.delete()
   D: removeElement()
   Ans:

6. **Which of the following is NOT a Map implementation?**
   A: HashMap
   B: TreeMap
   C: LinkedList
   D: Hashtable

   Ans:

7. **What does the Comparable interface allow you to do?**
   A: Define natural ordering for objects
   B: Compare objects for equality
   C: Synchronize object access
   D: Convert objects to strings

   Ans:

8. **Which utility class provides static methods for collection algorithms (e.g., sort, search)?**
   A: Collections
   B: Arrays
   C: ListUtils
   D: CollectionHelper

   Ans:

9. **Which of these legacy classes is part of the Collections Framework?**
   A: Vector
   B: Stack
   C: Hashtable
   D: All of the above

   Ans:

Short Answer Questions

1. Explain the difference between ArrayList and LinkedList.

2. What is the role of a Comparator in the Collections Framework?

3. Describe fail-fast behavior in Java collections.

4. How do you convert an array to a collection in Java?

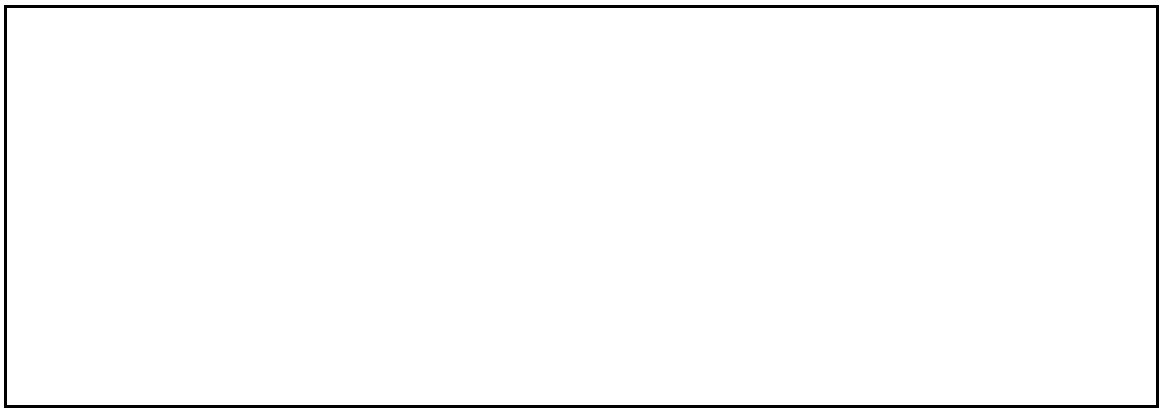
5. What is the significance of generics in collections?

Coding/Design Challenges

1. **Iterator Traversal:**
   Write a code snippet that uses an Iterator to traverse an ArrayList of custom objects

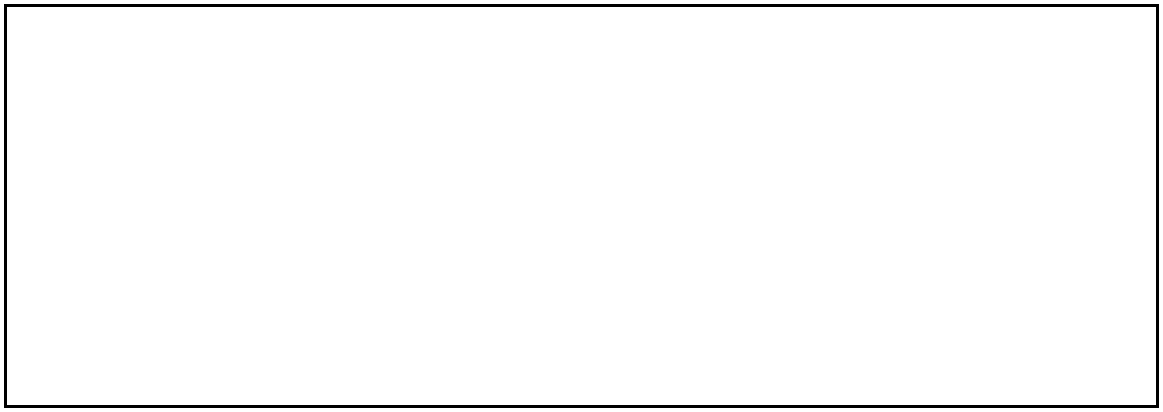
   (e.g., Student with fields name and id).

2. **Custom Sorting:**
   Design a Comparator to sort a list of `Student` objects by name, then by id if names are



   identical.

3. **Synchronization Challenge:**
   Write a thread-safe method that iterates over a shared `ArrayList` of integers and sums



   them. Use synchronization to avoid ConcurrentModificationException.

Interview Questions

1. What are the advantages of the Collections Framework in Java?

2. How does the fail-fast mechanism work in Java collections?

3. Can you explain the differences between ArrayList and Vector?

4. How would you store user-defined objects in a collection and sort them?

5. Describe how generics improve the safety and usability of collections.

Module 2: String Handling

MCQs

1. **Which of the following is immutable?**
   A: String
   B: StringBuffer
   C: StringBuilder
   D: All are mutable

   Ans:

2. **What does the length() method of a String return?**
   A: The capacity of the string
   B: The number of characters
   C: The size of the underlying array
   D: The index of the last character

   Ans:

3. **Which class is used for mutable string manipulation in a multi-threaded environment?**
   A: String
   B: StringBuffer
   C: StringBuilder
   D: None of the above

   Ans:

4. **What is the result of `"Hello".concat(" World")`?**
   A: "Hello World"
   B: "Hello"
   C: "World Hello"
   D: "HelloWorld"

   Ans:

5. **Which method is used to convert different data types into String?**
   A: toString()
   B: valueOf()
   C: parseString()
   D: convert()

   Ans:

6. **How do you extract a character at a specific index from a string?**
   A: getCharAt(index)
   B: charAt(index)
   C: substring(index)
   D: indexOf(index)

   Ans:

7. **What is the output of `"abc".compareTo("abd")`?**
   A: 0
   B: A negative value
   C: A positive value
   D: Depends on the JVM

   Ans:

8. **Which method would you use to join an array of strings?**
   A: join()
   B: combine()

C: concat()
D: merge()

Ans:

9. **Which of the following classes is not thread-safe?**
   A: StringBuffer
   B: StringBuilder
   C: String
   D: Both A and C

   Ans:

10. **What happens if you try to modify a String after it's created?**
    A: The original String is modified
    B: A new String object is created
    C: An error is thrown
    D: Nothing happens

    Ans:

Short Answer Questions

1. Differentiate between StringBuffer and StringBuilder.

2. How does the `equals()` method differ from == when comparing strings?

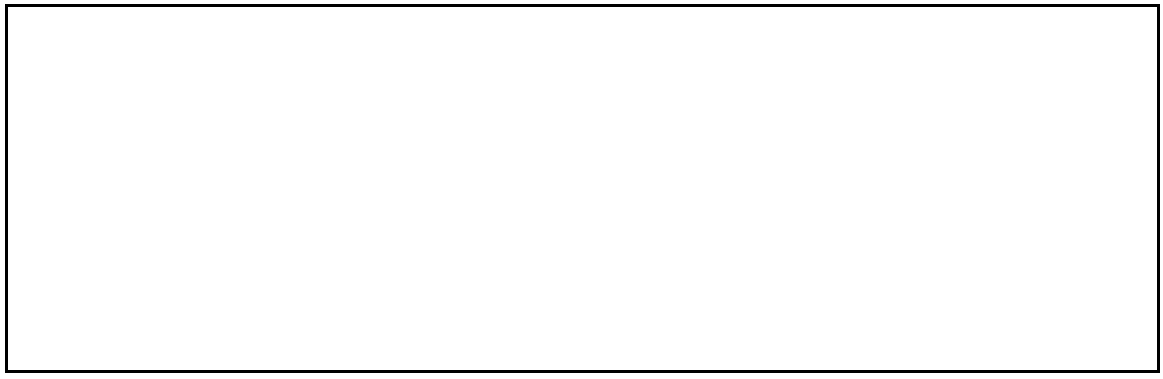
3. Explain how the immutability of String benefits security and performance.

4. How can you convert a primitive type to a String?

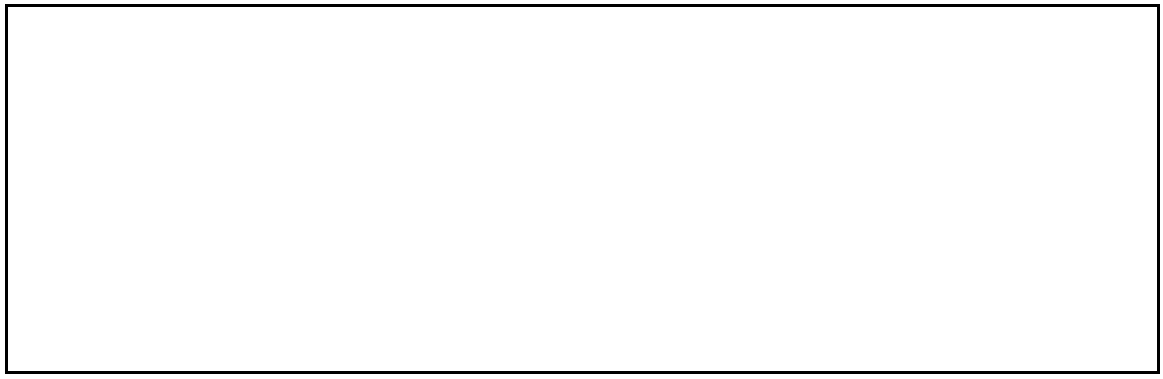
5. What is substring extraction and how is it performed?

Coding/Design Challenges

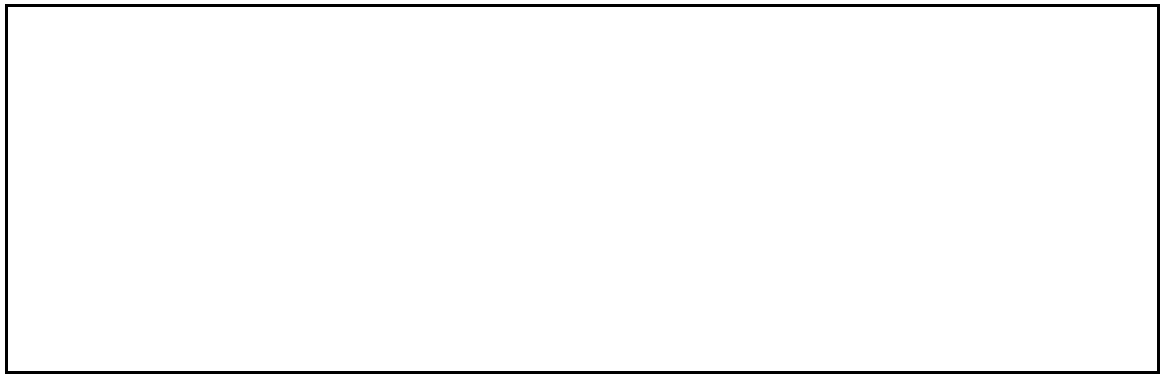
1. **Character Extraction:**
   Write a Java program that takes a string and prints each character on a new line.



2. **String Joining:**
   Design a method that joins an array of strings into a single string separated by commas.



3. **Mutable String Manipulation:**
   Using StringBuilder, write a function that reverses a given string without using the reverse() method.



Interview Questions

1. Why is String immutable in Java and what are the benefits?

2. How do you perform data conversion from various types to String in Java?

3. What are the differences between StringBuffer and StringBuilder?

4. How can you compare two strings lexicographically?

5. What is the significance of the string pool?

Module 3: Introducing Swing

MCQs

1. **Which package contains the Swing components?**
   A: java.awt
   B: javax.swing
   C: java.swing
   D: java.awt.swing

   Ans:

2. **What does MVC stand for in Swing?**
   A: Model-View-Control
   B: Model-View-Controller
   C: Main-View-Controller
   D: Model-Variable-Component

   Ans:

3. **Which of these is a Swing container?**
   A: JButton
   B: JPanel
   C: JLabel
   D: JTextField

   Ans:

4. **Which method is used to add an ActionListener to a JButton?**
   A: addListener()
   B: addActionListener()
   C: onClick()
   D: setActionListener()

   Ans:

5. **Which Swing component is used to create a window?**
   A: JFrame
   B: JPanel
   C: JDialog
   D: JWindow

   Ans:

6. **What is the default layout manager for a JFrame's content pane?**
   A: BorderLayout
   B: FlowLayout
   C: GridLayout
   D: BoxLayout

   Ans:

7. **Which class is used for thread-safe updates in Swing?**
   A: SwingWorker
   B: Thread
   C: Runnable
   D: Executor

   Ans:

8. **How do you repaint a Swing component?**
   A: refresh()
   B: redraw()
   C: repaint()
   D: update()

   Ans:

9. **What does the `pack()` method do in a JFrame?**
   A: Sets the frame size to 0
   B: Sizes the frame so that all its contents are at or above their preferred sizes
   C: Minimizes the frame
   D: Centers the frame on the screen

   Ans:

10. **Which layout manager arranges components in a grid?**
    A: FlowLayout
    B: BorderLayout
    C: GridLayout
    D: CardLayout

    Ans:

Short Answer Questions

1. Explain the Model-View-Controller (MVC) pattern in Swing.

2. What is the role of the Event Dispatch Thread (EDT) in Swing?

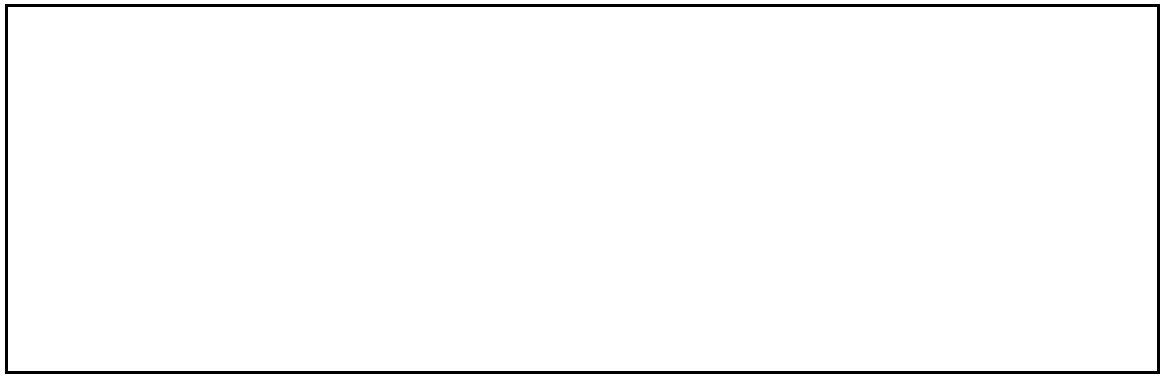
3. How can you perform custom painting in Swing?

4. Differentiate between JFrame and JDialog.

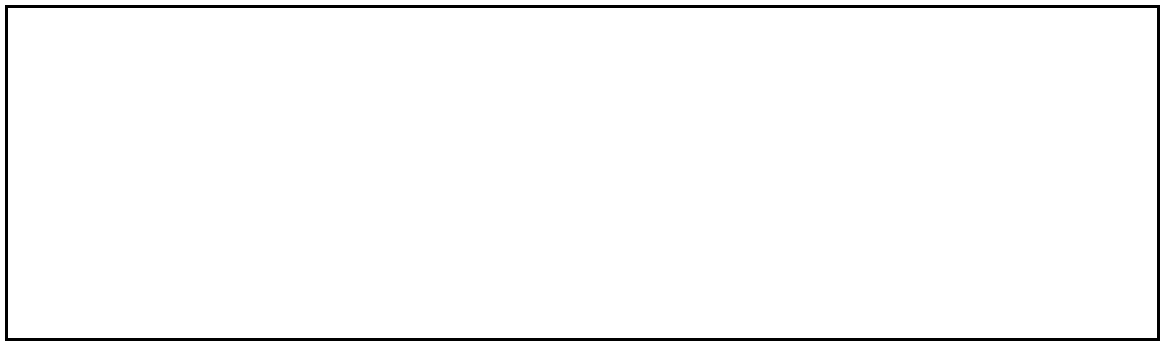
5. What is a SwingWorker and when should it be used?

Coding/Design Challenges

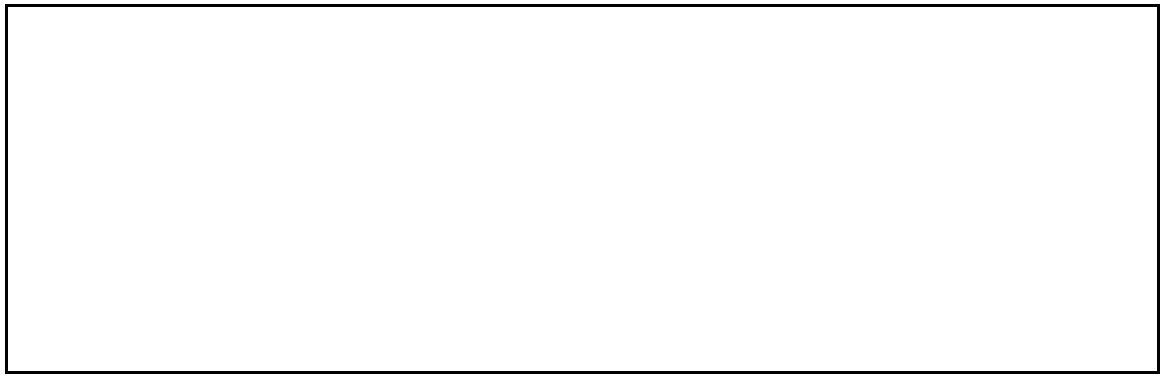
1. **Simple Swing Application:**
   Create a simple Swing application with a JFrame and a JButton that shows a dialog when clicked.



2. **Custom Component Painting:**
   Write a custom JPanel that draws a filled circle.



3. **SwingWorker Example:**
   Design a SwingWorker to simulate a background task that updates a progress bar.



Interview Questions

1. What is the MVC pattern, and why is it important in Swing?

2. Explain the role of the Event Dispatch Thread (EDT) in Swing.

3. How would you update a Swing component from a background thread?

4. What are some common pitfalls in Swing application development?

5. How can you improve the performance of a Swing application?

Module 4: Introducing Servlets & JSP

MCQs

1. **Which method in HttpServlet handles GET requests?**
   A: doGet()
   B: doPost()
   C: service()
   D: init()

   Ans:

2. **Which descriptor file configures servlets in a web application?**
   A: web.xml
   B: servlet.xml
   C: app.xml
   D: server.xml

   Ans:

3. **What is the primary purpose of a Servlet?**
   A: To generate dynamic web content
   B: To store files
   C: To manage databases
   D: To handle file I/O

   Ans:

4. **Which package contains HttpServlet?**
   A: javax.servlet.http
   B: java.servlet.http
   C: javax.http.servlet
   D: java.servlet

   Ans:

5. **What does JSP stand for?**
   A: Java Standard Pages
   B: Java Server Pages
   C: JavaScript Pages
   D: Java Servlet Pages

   Ans:

6. **How is session tracking typically achieved in servlets?**
   A: Using cookies or URL rewriting
   B: Using file I/O
   C: Through the database
   D: Using JSP tags

   Ans:

7. **Which HTTP method is idempotent?**
   A: POST
   B: GET
   C: PUT
   D: DELETE

   Ans:

8. **What is the role of the ServletContext?**
   A: To hold session-specific data
   B: To manage application-wide parameters and resources
   C: To handle individual user requests
   D: To store cookies

   Ans:

9. **Which of these is used for forwarding requests in a servlet?**
   A: RequestDispatcher
   B: ResponseDispatcher
   C: ServletForwarder
   D: ContextForwarder

   Ans:

10. **Which of the following is a JSP scripting element used to output values?**
    A: <% ... %>
    B: <%= ... %>
    C: <%! ... %>
    D: <%-- ... --%>

    Ans:

Short Answer Questions

1. Describe the servlet life cycle.

2. How does session tracking work using cookies?

3. What is the main advantage of using JSP over servlets for generating dynamic content?

4. Explain the difference between forwarding and redirection in servlets.

5. How can you access initialization parameters in a servlet?

Coding/Design Challenges

1. **Simple Servlet Creation:**
   Write a simple servlet that handles a GET request and responds with "Hello, World!" in HTML.

2. **Request Dispatching:**
   Design a servlet that forwards a request to a JSP page named `result.jsp`.

3. **Session Handling:**
   Write a servlet snippet that retrieves a session attribute "username" and prints it; if not present, set it to "guest".

Interview Questions

1. What are the advantages of using servlets over CGI scripts?

2. How do you maintain session state in a web application?

3. Explain the difference between GET and POST methods in HTTP requests.

4. What is the role of web.xml in a servlet-based application?

5. How would you handle errors in a servlet?

Module 5: JDBC Objects

MCQs

1. **What does JDBC stand for?**
   A: Java DataBase Connectivity
   B: Java Direct Base Connection
   C: Java Distributed Base Connection
   D: Java DataBase Component

   Ans:

2. **Which type of JDBC driver is 100% pure Java?**
   A: Type 1
   B: Type 2
   C: Type 3
   D: Type 4

   Ans:

3. **Which package contains JDBC classes?**
   A: java.jdbc
   B: java.sql
   C: javax.sql
   D: java.database

   Ans:

4. **What is the purpose of the DriverManager class?**
   A: To manage a list of JDBC drivers
   B: To close connections
   C: To execute SQL queries
   D: To handle transactions

   Ans:

5. **Which method is used to create a database connection?**
   A: connect()
   B: getConnection()

C: createConnection()
D: open()

Ans:

6. **Which interface represents a result set in JDBC?**
   A: Statement
   B: ResultSet
   C: Connection
   D: Driver

   Ans:

7. **What is the default auto-commit mode in JDBC?**
   A: true
   B: false
   C: depends on the driver
   D: not applicable

   Ans:

8. **Which exception is commonly caught when handling JDBC errors?**
   A: IOException
   B: SQLException
   C: ClassNotFoundException
   D: NullPointerException

   Ans:

9. **Which type of SQL operation would you execute using executeUpdate()?**
   A: SELECT
   B: INSERT, UPDATE, DELETE
   C: CREATE
   D: DROP

   Ans:

10. **Which JDBC driver type is recommended for enterprise applications?**
    A: Type 1
    B: Type 2
    C: Type 3
    D: Type 4

Ans:

Short Answer Questions

1. What are the four types of JDBC drivers?

2. Describe the steps to establish a JDBC connection.

3. How do you manage transactions in JDBC?

4. What is a PreparedStatement and why is it preferred over Statement?

5. Explain the role of ResultSet in JDBC.

Coding/Design Challenges

1. **Database Connection:**
   Write a Java program to establish a connection to a MySQL database and print a success message.

2. **Executing an Update Query:**
   Design a code snippet that updates a student's grade in the database using a PreparedStatement.

3. **Query and Display:**
   Write a code snippet that executes a SELECT query and prints out the results from a table named "employees".

Interview Questions

1. What are the advantages of using PreparedStatement over Statement in JDBC?

2. How do you handle transactions in JDBC?

3. What is connection pooling, and why is it used?

4. How can you prevent SQL injection in a JDBC application?

5. Describe the role of DriverManager in JDBC.

Overview & Course Outcomes

Course Skill Set:

1. Apply appropriate collection class/interface to solve a given problem.
2. Demonstrate the concepts of string operations in Java.
3. Apply Swing concepts to build Java desktop applications.
4. Develop web-based applications using Java servlets and JSP.
5. Use JDBC to build database applications.

Project Ideas:

- **Swing Calculator:** Build a calculator using Swing components (CO3).
- **Student Registration App:** Develop a registration web app using Servlets/JSP (CO4).

Additional Resources:

1. **Official Java Documentation:**
    - Java SE Documentation
    - Collections Framework Tutorial
2. **Books & Articles:**
    - *Effective Java* by Joshua Bloch
    - *Java_The_Complete_Reference_13th_Ed_-_Herbert_Schildt.pdf*
    - Baeldung Articles on Java Collections, String Handling, and JDBC
3. **Swing Resources:**
    - Oracle Swing Tutorial
    - TutorialsPoint – Swing Tutorial
4. **Servlets & JSP:**
    - Oracle Servlets and JSP Documentation
    - JSP Tutorial on TutorialsPoint
5. **JDBC Resources:**
    - JDBC Tutorial – Oracle
    - JDBC Best Practices – Baeldung
6. **Youtube Videos:**
    - **OOP 1 | Introduction & Concepts - Classes, Objects, Constructors, Keywords**
    - **Master Java Collections Framework in 3 Hours 🔥🔥 | Full Course in Depth | For DSA | Java Tutorial**
    - **Introduction to Swing in Java | Free Java Course**
    - **Servlet & JSP Tutorial | Full Course**
    - **JDBC in Java | Java Database Connectivity | Java Tutorial | Great Learning**