

# Advance Java Lab Manual

## BIS402

## IPCC

## 4th semester

1. **Implement a java program to demonstrate creating an ArrayList, adding elements, removing elements, sorting elements of ArrayList. Also illustrate the use of toArray() method.**

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;

public class ArrayListExample {
    public static void main(String[] args) {
        // Create an ArrayList
        ArrayList<Integer> arrayList = new ArrayList<>();

        // Add elements to the ArrayList
        arrayList.add(5);
        arrayList.add(2);
        arrayList.add(8);
        arrayList.add(1);
        arrayList.add(9);

        System.out.println("Initial ArrayList: " + arrayList);

        // Remove an element from the ArrayList
        arrayList.remove(2);
        System.out.println("ArrayList after removing an element: " + arrayList);

        // Sort the ArrayList
        Collections.sort(arrayList);
        System.out.println("Sorted ArrayList: " + arrayList);

        // Convert the ArrayList to an array
        Integer[] array = arrayList.toArray(new Integer[0]);
        System.out.println("Array: " + Arrays.toString(array));
    }
}
```

Output:

```
Initial ArrayList: [5, 2, 8, 1, 9]
ArrayList after removing an element: [5, 2, 1, 9]
Sorted ArrayList: [1, 2, 5, 9]
Array: [1, 2, 5, 9]
```

2. **Develop a program to read random numbers between a given range that are multiples of 2 and 5, sort the numbers according to tens place using comparator.**

```
import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;
```

```
import java.util.Random;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<>();
        Random random = new Random();

        // Generate random numbers between 10 and 100 that are multiples of 2 and 5
        while (numbers.size() < 10) {
            int number = random.nextInt(41) * 5;
            if (number % 2 == 0 && number >= 10 && number <= 100) {
                numbers.add(number);
            }
        }

        // Sort the numbers according to the tens place
        Collections.sort(numbers, new Comparator<Integer>() {
            @Override
            public int compare(Integer o1, Integer o2) {
                return Integer.compare(o1 / 10, o2 / 10);
            }
        });

        // Print the sorted numbers
        for (int number : numbers) {
            System.out.println(number);
        }
    }
}
```

**3. Implement a java program to illustrate storing user defined classes in collection.**

```
import java.util.ArrayList;
import java.util.List;

class Student {
    private String name;
    private int age;

    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    @Override
    public String toString() {
        return "Student{" +
            "name=" + name + "\" +
            ", age=" + age +
            "\"";
    }
}
```

```
public class Main {  
    public static void main(String[] args) { List<Student> students = new ArrayList<>();  
  
    // Add students to the list students.add(new Student("Alice", 20));  
    students.add(new Student("Bob", 22));  
    students.add(new Student("Charlie", 23));  
  
    // Print the students  
    for (Student student : students) {  
        System.out.println(student);  
    }  
}  
}
```

**Output:**

```
Student{name='Alice', age=20}  
Student{name='Bob', age=22}  
Student{name='Charlie', age=23}
```

**4. Implement a java program to illustrate the use of different types of string class constructors.**

```
public class Main {  
    public static void main(String[] args) {  
        // 1. Default constructor: Creates a new empty string  
        String str1 = new String();  
        System.out.println("Default constructor: " + str1);  
  
        // 2. Constructor with a string argument: Creates a new string with the specified value  
        String str2 = new String("Hello, World!");  
        System.out.println("Constructor with a string argument: " + str2);  
  
        // 3. Constructor with a character array: Creates a new string with the specified character  
        array  
        char[] charArray = {'H', 'e', 'l', 'l', 'o'};  
        String str3 = new String(charArray);  
        System.out.println("Constructor with a character array: " + str3);  
    }  
}
```

// 4. Constructor with a character array and offset: Creates a new string with the specified character array and offset

```
char[] charArrayOffset = {'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd'};  
String str4 = new String(charArrayOffset, 7, 5);  
System.out.println("Constructor with a character array and offset: " + str4);
```

// 5. Constructor with a byte array: Creates a new string with the specified byte array

```
byte[] byteArray = {72, 101, 108, 108, 111};  
String str5 = new String(byteArray);  
System.out.println("Constructor with a byte array: " + str5);
```

// 6. Constructor with a byte array and charset: Creates a new string with the specified byte array and charset

```
byte[] byteArrayCharset = {72, 101, 108, 108, 111};  
String str6 = new String(byteArrayCharset, "UTF-8");  
System.out.println("Constructor with a byte array and charset: " + str6);
```

// 7. Constructor with a byte array, offset, and length: Creates a new string with the specified byte array, offset, and length

```
byte[] byteArrayOffset = {72, 101, 108, 108, 111, 32, 87, 111, 114, 108, 100};  
String str7 = new String(byteArrayOffset, 7, 5);  
System.out.println("Constructor with a byte array, offset, and length: " + str7);
```

// 8. Constructor with a byte array, offset, length, and charset: Creates a new string with the specified byte array, offset, length, and charset

```
byte[] byteArrayOffsetCharset = {72, 101, 108, 108, 111, 32, 87, 111, 114, 108, 100};  
String str8 = new String(byteArrayOffsetCharset, 7, 5, "UTF-8");  
System.out.println("Constructor with a byte array, offset, length, and charset: " + str8);  
}  
}
```

Output:

Default constructor:

Constructor with a string argument: Hello, World!

Constructor with a character array: Hello

Constructor with a character array and offset: World

Constructor with a byte array: Hello

Constructor with a byte array and charset: Hello

Constructor with a byte array, offset, and length: World

Constructor with a byte array, offset, length, and charset: World

**5. Implement a java program to illustrate the use of different types of character extraction, string comparison, string search and string modification methods.**

```
public class StringMethodsExample {  
    public static void main(String[] args) {  
        String str = "Hello World!";  
  
        // Character extraction methods  
        System.out.println("Character at index 0: " + str.charAt(0));  
        System.out.println("First occurrence of 'l': " + str.indexOf('l'));  
        System.out.println("Last occurrence of 'l': " + str.lastIndexOf('l'));  
        System.out.println("Substring from index 0 to 5: " + str.substring(0, 5));  
  
        // String comparison methods  
        String str1 = "Hello World!";  
        String str2 = "hello world!";  
        System.out.println("Equal ignoring case: " + str1.equalsIgnoreCase(str2));  
        System.out.println("Starts with 'Hello': " + str1.startsWith("Hello"));  
        System.out.println("Ends with '!': " + str1.endsWith("!"));  
  
        // String search methods  
        System.out.println("Contains 'World': " + str1.contains("World"));  
        System.out.println("Index of '!': " + str1.indexOf('!'));  
        System.out.println("Last index of 'l': " + str1.lastIndexOf('l'));  
  
        // String modification methods  
        System.out.println("Uppercase: " + str1.toUpperCase());  
        System.out.println("Lowercase: " + str1.toLowerCase());  
    }  
}
```

```
        System.out.println("Trim: " + " Hello World! ".trim());  
        System.out.println("Replace 'World' with 'Java': " + str1.replace("World", "Java"));  
        System.out.println("Concatenate ' and goodbye': " + str1.concat(" and goodbye"));  
    }  
}
```

Output:

Character at index 0: H

First occurrence of 'l': 2

Last occurrence of 'l': 9

Substring from index 0 to 5: Hello

Equal ignoring case: true

Starts with 'Hello': true

Ends with 'l': true

Contains 'World': true

Index of 'l': 11

Last index of 'l': 9

Uppercase: HELLO WORLD!

Lowercase: hello world!

Trim: Hello World!

Replace 'World' with 'Java': Hello Java!

Concatenate ' and goodbye': Hello World! and goodbye

## 6. Implement a java program to illustrate the use of different types of StringBuffer methods

```
public class StringBufferMethodsExample {  
    public static void main(String[] args) {  
        StringBuffer sb = new StringBuffer("Hello World!");  
  
        // Length method  
        System.out.println("Length: " + sb.length());  
    }  
}
```



```
// Capacity method
System.out.println("Capacity: " + sb.capacity());

// Set length method
sb.setLength(10);
System.out.println("Set length to 10: " + sb);

// Ensure capacity method
sb.ensureCapacity(20);
System.out.println("Ensure capacity of 20: " + sb.capacity());

// Insert method
sb.insert(5, " Java");
System.out.println("Insert ' Java' at index 5: " + sb);

// Delete method
sb.delete(5, 10);
System.out.println("Delete from index 5 to 10: " + sb);

// Reverse method
sb.reverse();
System.out.println("Reverse: " + sb);

// Append method
sb.append(" and goodbye");
System.out.println("Append ' and goodbye': " + sb);

// Insert method (with object)
sb.insert(0, "Hello ");
System.out.println("Insert 'Hello ' at index 0: " + sb);

// DeleteCharAt method
```

```
        sb.deleteCharAt(5);  
        System.out.println("Delete char at index 5: " + sb);  
  
        // Replace method  
        sb.replace(0, 5, "Hi");  
        System.out.println("Replace 'Hello ' with 'Hi': " + sb);  
    }  
}
```

Output:

Length: 11

Capacity: 16

Set length to 10: Hello W

Ensure capacity of 20: 20

Insert Java at index 5: Hello JavaWorld!

Delete from index 5 to 10: Hello !

Reverse: !dlroW olleH

Append and goodbye: !dlroW olleH and goodbye

Insert Hello at index 0: Hello !dlroW olleH and goodbye

Delete char at index 5: Hello World and goodbye

Replace Hello with Hi: Hi World and goodbye

**7. Demonstrate a swing event handling application that creates 2 buttons Alpha and Beta and displays the text “Alpha pressed” when alpha button is clicked and “Beta pressed” when beta button is clicked.**

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class SwingEventHandlingExample {  
    public static void main(String[] args) {  
        // Create a new JFrame
```

```
JFrame frame = new JFrame("Swing Event Handling Example");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Create a new JPanel
JPanel panel = new JPanel();

// Create two JButtons
JButton alphaButton = new JButton("Alpha");
JButton betaButton = new JButton("Beta");

// Create an ActionListener for the Alpha button
ActionListener alphaListener = new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(frame, "Alpha pressed");
    }
};

// Create an ActionListener for the Beta button
ActionListener betaListener = new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(frame, "Beta pressed");
    }
};

// Add ActionListeners to the buttons
alphaButton.addActionListener(alphaListener);
betaButton.addActionListener(betaListener);

// Add the buttons to the panel
panel.add(alphaButton);
```

```
panel.add(betaButton);

// Add the panel to the frame
frame.getContentPane().add(panel, BorderLayout.CENTER);

// Set the frame size and make it visible
frame.setSize(300, 200);
frame.setVisible(true);
}
}
```

**8. A program to display greeting message on the browser “Hello UserName”, “How Are You?”, accept username from the client using servlet.**

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class GreetingServlet extends HttpServlet {
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Get the user's name from the request parameter
        String userName = request.getParameter("userName");

        // Set the content type of the response
        response.setContentType("text/html");

        // Get the print writer to write the response
        PrintWriter out = response.getWriter();

        // Write the HTML response
        out.println("<html>");
```

```
        out.println("<head><title>Greeting Servlet</title></head>");
        out.println("<body>");
        out.println("<h1>Hello " + userName + "!</h1>");
        out.println("<h2>How are you?</h2>");
        out.println("</body>");
        out.println("</html>");

        // Close the print writer
        out.close();
    }
}
```

Output:

Hello John!

How are you?

### 9. A servlet program to display the name, USN, and total marks by accepting student details

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class StudentMarksServlet extends HttpServlet {
    String message, USN;
    Connection connect;
    Statement stmt = null;
    ResultSet rs = null;

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            String url = "jdbc:odbc:NEO";
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
        connect = DriverManager.getConnection(url, " ", " ");
        message = "Memorandum of Marks";
    } catch (ClassNotFoundException cnfex) {
        cnfex.printStackTrace();
    } catch (SQLException sqllex) {
        sqllex.printStackTrace();
    } catch (Exception excp) {
        excp.printStackTrace();
    }
}

USN = request.getParameter("usn");
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head>");
out.println("</head>");
out.println("<body bgcolor=cyan>");
out.println("<center>");
out.println("<h1>" + message + "</h1>\n");

try {
    stmt = connect.createStatement();
    String query = new String("SELECT * FROM StudentMarks WHERE USN=" + USN);
    rs = stmt.executeQuery(query);
    boolean b = rs.next();
    out.println("USN: " + rs.getString(1));
    out.println("<br>" + "Name: " + rs.getString(2));
    out.println("<br>" + "Total Marks: " + rs.getInt(3));
} catch (SQLException ex) {
    out.println("error in connection");
} finally {
    try {
```

```
        if (rs != null)
            rs.close();
        if (stmt != null)
            stmt.close();
        if (connect != null)
            connect.close();
    } catch (SQLException e) {
    }
}

out.println("</center>");
out.println("</body></html>");
}
}
```

**10. A Java program to create and read the cookie for the given cookie name as “EMPID” and its value as “AN2356”.**

```
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import javax.servlet.*;
import javax.servlet.http.*;

public class CookieExample extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Create a new cookie with the name "EMPID" and the value "AN2356"
        Cookie cookie = new Cookie("EMPID", "AN2356");

        // Set the maximum age of the cookie to 60 seconds
        cookie.setMaxAge(60);

        // Add the cookie to the response
```

```
response.addCookie(cookie);

// Read the cookie from the response
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for (Cookie c : cookies) {
        if (c.getName().equals("EMPID")) {
            System.out.println("Cookie Name: " + c.getName());
            System.out.println("Cookie Value: " + c.getValue());
        }
    }
}
}
```

output:

Cookie Name: EMPID

Cookie Value: AN2356

**11. Write a JAVA Program to insert data into Student DATA BASE and retrieve info based on particular queries(For example update, delete, search etc...).**

```
import java.sql.*;

public class StudentDatabase {
    private static final String DB_URL = "jdbc:mysql://localhost:3306/studentdb";
    private static final String DB_USER = "root";
    private static final String DB_PASSWORD = "password";

    public static void main(String[] args) {
        try {
            // Step 1: Register JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");
```



```
// Step 2: Open a connection
Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);

// Step 3: Execute a query
Statement stmt = conn.createStatement();

// Insert a new student
String sql = "INSERT INTO students (id, name, age) VALUES (1, 'John Doe', 20)";
stmt.executeUpdate(sql);
System.out.println("Student added successfully.");

// Update a student's age
sql = "UPDATE students SET age = 21 WHERE id = 1";
stmt.executeUpdate(sql);
System.out.println("Student updated successfully.");

// Search for a student by name
sql = "SELECT * FROM students WHERE name = 'John Doe'";
ResultSet rs = stmt.executeQuery(sql);
while (rs.next()) {
    System.out.println("ID: " + rs.getInt("id"));
}
```

## 12. A program to design the Login page and validating the USER\_ID and PASSWORD using JSP and DataBase.

### Login.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <form method="post" action="login">
        User ID: <input type="text" name="userid"><br>
```

```
        Password: <input type="password" name="password"><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

**LoginServlet.java:**

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class LoginServlet extends HttpServlet {
    private static final String DB_URL = "jdbc:mysql://localhost:3306/userdb";
    private static final String DB_USER = "root";
    private static final String DB_PASSWORD = "password";

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String userid = request.getParameter("userid");
        String password = request.getParameter("password");

        try {
            // Step 1: Register JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Step 2: Open a connection
            Connection conn = DriverManager.getConnection(DB_URL, DB_USER,
                DB_PASSWORD);

            // Step 3: Execute a query
            Statement stmt = conn.createStatement();

            // Validate user ID and password
            String sql = "SELECT * FROM users WHERE userid = '" + userid + "' AND password = '" +
                password + "'";
            ResultSet rs = stmt.executeQuery(sql);
            if (rs.next()) {
                // User is valid, redirect to welcome page
                RequestDispatcher rd = request.getRequestDispatcher("welcome.jsp");
                rd.forward(request, response);
            } else {
                // User is not valid, show error message
                request.setAttribute("error", "Invalid user ID or password");
                RequestDispatcher rd = request.getRequestDispatcher("login.jsp");
                rd.forward(request, response);
            }

            // Step 4: Close the connection
            conn.close();
        }
    }
}
```

```
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}
```

**Welcome.jsp:**

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Welcome</title>
</head>
<body>
    <h1>Welcome, ${userid}!</h1>
    <p><a href="logout">Logout</a></p>
</body>
</html>
```

**LogoutServlet.java:**

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class LogoutServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Clear session
        request.getSession().invalidate();

        // Redirect to login page
        response.sendRedirect("login.jsp");
    }
}
```