

# Information Booklet cum Syllabus

Of

## Java Programming

Revision-I



July 2022

**National Institute of Electronics and Information Technology**

An Autonomous Scientific Society under  
Ministry of Electronics and Information Technology, Government of India

NIELIT Gorakhpur  
Deoria Road  
Gorakhpur (U.P.) -273010

NIELIT Extension Centre Lucknow  
NIELIT Lucknow  
Sumit Complex, A-1/9, Vibhuti Khand,  
Gomti Nagar, Lucknow – 226010

<b>CONTENTS</b>		
<b>Sl. No.</b>	<b>Title</b>	<b>Page No.</b>
1.	About the course	02
2.	NIELIT	02
3.	Objective of Course	02
4.	Job Role of Course	02
5.	Eligibility	03
6.	Duration of Course	03
7.	Course Details	03
8.	Reference Books / Study Material	08
9.	Practical Assignments	08

## 1. **About Course**

The Java is one of the most popular programming languages because of its versatile and compatible nature. It can be used for Application development, Web development and Mobile Apps development. This Java course is designed to cover the fundamentals of Java programming including Object Oriented Programming (OOP) concepts. The learners of the course will be able to get extensive and deep knowledge of Core Java. The course includes comprehensive materials with programming exercises.

## 2. **NIELIT**

National Institute of Electronics and Information Technology, NIELIT, (Erstwhile DOEACC Society) is an autonomous scientific society of the Ministry of Electronics & Information Technology, Government of India. The Society is registered under the Societies Registration Act, 1860. NIELIT was set up to carry out Human Resource Development and related activities in the area of Information, Electronics & Communications Technology (IECT). NIELIT is engaged both in Formal & Non-Formal Education in the areas of IECT besides development of industry-oriented quality education and training programs in the state-of-the-art areas. NIELIT has endeavored to establish standards to be the country's premier institution for Examination and Certification in the field of IECT. It is also one of the National Examination Body, which accredits institutes/organizations for conducting courses in IT and Electronics in the non-formal sector.

## 3. **Objective of Course**

This objective of this course is to introduce the learner to the fundamentals of java programming. It starts with the basics, such as how to compile and run a Java program, and then covers OOPs concept, Array, Strings, Multithreading, I/O operations, Exception Handling, Swing and database connectivity.

After completing this course, the learner will be able to:

- Understand the programming of Java.
- Write, compile, run and test object-oriented java programs.
- Create statements, functions, loops to process information and solve problems.
- Understand the OOPs concepts
- Understand the file-based I/O operations.
- Understand the multithreading and Swing in Java.
- Understand the database connectivity in Java.

## 4. **Job Roles of Course**

After successful completion of the course the candidates shall be employed in the industries for following occupations:

- Java Programmer
- Java Developer
- Java Freelancer

## 5. **Eligibility**

Open to All.

## 6. Total duration of the Course

80 Hours (Theory: 32 hours, Practical/Tutorial: 48 hours)

## 7. Course Details

### 7.1. Course Outline and Objective of Each Unit

S. No.	Unit Name	Duration (Theory) in Hours	Duration (Practical) in Hours	Total Learning Hours	Learning Objectives
1.	Introduction to Java and Its Fundamentals	05	05	10	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Install and set the path of Java.</li> <li>• Create, Compile, Run the Java Programs</li> <li>• Familiar with fundamentals of Java like datatype, variables, tokens, operators etc.</li> </ul>
2.	Control and Looping Statments	02	04	06	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Know about the use of control statements and looping statements in Java Programming.</li> </ul>
3.	OOP's concept – Class, Object, Inheritance, Polymorphism, Abstraction, Encapsulation	06	06	12	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Understand and implementation of OOPs concepts - Class, Object, Inheritance, Polymorphism, Abstraction, Encapsulation</li> </ul>
4.	Packages	02	02	04	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Understand the concept of Packages in Java.</li> </ul>

					<ul style="list-style-type: none"> <li>• Use Built-in and customize Packages.</li> </ul>
5.	Arrays and String	03	05	08	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Know the concept of Arrays and String.</li> <li>• Manipulation and operations in Arrays and String.</li> </ul>
6.	Exception Handling	02	04	06	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Handle exception situations.</li> <li>• Use of try and catch statements in Java.</li> </ul>
7.	Multithreading	02	04	06	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Use of Threads in Java.</li> <li>• Life cycle of Thread.</li> <li>• Know Multithreading and its method.</li> </ul>
8.	Input/ Output Operations	01	03	04	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Work with input and output operations.</li> <li>• Work with reading/writing operations into files.</li> </ul>
9.	Event Handling and Swing	04	06	10	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Handle various events in Java like Mouse movements, Button clicks etc.</li> <li>• Develop Graphical User Interface in Java.</li> <li>• GUI Components – JButton, JCheckBox, JRadioButton,</li> </ul>

					JMenu, JPanel, JPasswordField etc.
10.	Introduction to SQL, Database connectivity	05	09	14	<p>After completing this unit, Learner will be able to</p> <ul style="list-style-type: none"> <li>• Understand the concept of Database.</li> <li>• Familiar with RDBMS technology.</li> <li>• Know about the SQL Statements.</li> <li>• Connect database with Java</li> </ul>
<b>Grand Total</b>		<b>32</b>	<b>48</b>	<b>80</b>	

**7.2.Detailed Syllabus**

<b>Unit Name</b>	<b>Contents</b>	<b>Hrs.</b>
Introduction to Java and Its Fundamentals	<ul style="list-style-type: none"> <li>• What is Java</li> <li>• History of Java</li> <li>• Features of Java</li> <li>• Comparison in Java with C and C++</li> <li>• JDK, JRE &amp; JVM</li> <li>• Installation of Java</li> <li>• How to Set Path?</li> <li>• Creating First Java Program</li> <li>• Compile, Run and Explanation of first Java Program</li> <li>• Data Type</li> <li>• Variables</li> <li>• Operators</li> <li>• Keywords</li> <li>• Constants</li> <li>• Identifiers</li> <li>• Comments</li> <li>• Type Casting</li> </ul>	10
Control and Looping Statments	<ul style="list-style-type: none"> <li>• if statements</li> <li>• if-else statement and if-else-if ladder</li> <li>• Nested if statements</li> <li>• switch statements</li> <li>• while loop</li> <li>• do-while loop</li> <li>• for loop</li> <li>• for-each loop</li> <li>• break and continue</li> </ul>	06

<p>OOP's concept – Class, Object, Inheritance, Polymorphism, Abstraction, Encapsulation</p>	<ul style="list-style-type: none"> <li>• OOP concepts with real life example</li> <li>• Defining a Class</li> <li>• Field declaration</li> <li>• Creating Objects</li> <li>• Constructor</li> <li>• Type of constructor</li> <li>• Methods</li> <li>• static keyword</li> <li>• this keyword</li> <li>• Introduction to Inheritance</li> <li>• Types of Inheritance</li> <li>• Introduction to Polymorphism</li> <li>• Type of polymorphism</li> <li>• Difference between method overloading &amp; method overriding</li> <li>• Super keyword and it's usages</li> <li>• Final keyword and it's usages</li> <li>• Different way to achieve Abstraction in java Abstract class</li> <li>• Interface</li> <li>• Multiple inheritance by interface</li> <li>• Difference between abstract &amp; Interface</li> </ul>	<p>12</p>
<p>Packages</p>	<ul style="list-style-type: none"> <li>• Understanding Package</li> <li>• Using system packages</li> <li>• Creating package</li> <li>• Accessing a package</li> <li>• Using a package</li> <li>• Setting Class path</li> <li>• Reading Input from Keyboard</li> <li>• Access Modifiers</li> <li>• Implementation of Encapsulation</li> </ul>	<p>04</p>
<p>Arrays, Vector and String</p>	<ul style="list-style-type: none"> <li>• Introduction to Array</li> <li>• Advantages of Array</li> <li>• Creating an Array</li> <li>• Initializing &amp; Accessing Array</li> <li>• 1-d Arrays</li> <li>• 2-d Arrays</li> <li>• Jagged Arrays</li> <li>• Anonymous array</li> <li>• Operations on Arrays</li> <li>• Vector</li> <li>• What is String</li> <li>• String Class</li> <li>• Creating String Object</li> <li>• Operations on String</li> <li>• StringBuffer class and it's Methods Difference between String and StringBuffer class</li> <li>• StringBuilder Class and it's Methods</li> <li>• Difference between StringBuffer and StringBuilder</li> </ul>	<p>08</p>

Exception Handling	<ul style="list-style-type: none"> <li>• Introduction to Exception</li> <li>• Types of exception</li> <li>• Exception Class Hierarchy</li> <li>• Try &amp; Catch Blocks</li> <li>• Nested Try statements</li> <li>• Throw, throws and finally keyword</li> <li>• Creating Custom Exceptions</li> <li>• Difference between Final, Finally and Finalize</li> </ul>	06
Multithreading	<ul style="list-style-type: none"> <li>• Understanding Threads and process</li> <li>• Multiprocessing v/s Multithreading</li> <li>• How to create thread</li> <li>• Creating Child Threads and understanding context switching</li> <li>• Thread Life-Cycle</li> <li>• Thread Priorities</li> <li>• Main Thread (Default Java Thread)</li> <li>• Performing multiple job by multiple Thread</li> <li>• Sleeping a thread</li> <li>• Using thread methods</li> <li>• Daemon thread</li> </ul>	06
Input/ Output Operations	<ul style="list-style-type: none"> <li>• What is I/O</li> <li>• Why Need Streams</li> <li>• Byte Streams and Character Streams</li> <li>• Read/Write operations with file</li> <li>• Scanner Class</li> <li>• Input from keyboard by InputStreamReader</li> </ul>	04
Event Handling and Swing	<ul style="list-style-type: none"> <li>• Event Handling Mechanism</li> <li>• Event Classes, Event Listener Interfaces</li> <li>• Introduction to Swing</li> <li>• Difference between AWT and Swing</li> <li>• Hierarchy of java swing classes</li> <li>• JButton class</li> <li>• JLabel</li> <li>• JTextField &amp; JTextArea class</li> <li>• JPasswordField class</li> <li>• JRadioButton class</li> <li>• JComboBox class</li> <li>• JTable class</li> <li>• Displaying Image</li> <li>• Layout Managers</li> <li>• JMenu, JMenuItem and JMenuBar</li> </ul>	10
Introduction to SQL, Java Database connectivity	<ul style="list-style-type: none"> <li>• RDBMS Technology</li> <li>• Keys in database.</li> <li>• Introduction to SQL,</li> <li>• SQL statements – DDL and DML</li> <li>• Operators, Order By, Group By</li> <li>• Aggregate Function</li> <li>• SQL Join and its types, Sub query</li> <li>• Introduction to JDBC</li> <li>• Steps to connect to the database</li> </ul>	14



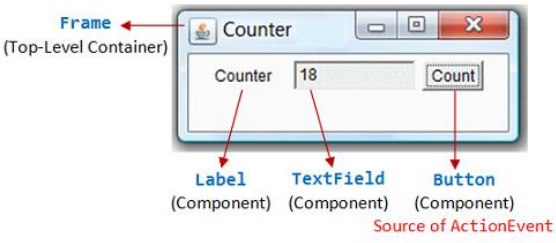
	<ul style="list-style-type: none"><li>• Types of JDBC Drivers</li><li>• Connectivity with MySQL</li></ul>	
<b>Total</b>		<b>80</b>

**8. Reference Books/Study Material**

- Java: The Complete Reference - McGraw Hill Education - Herbert Schildt
- Core Java: An Integrated Approach - Dreamtech Press - Dr. R. Nageswara Rao
- Programming with Java - McGraw-Hill - E. Balagurusamy
- Let Us Java - BPB Publications - Yashavant Kanetkar
- Beginning Programming with Java For Dummies – Wiley - Dr. Barry Burd
- Internet And Java Programming - New Age International - R Krishnamoorthy

## 9. Practical Assignments

<p><b><u>Assignment 1.</u></b></p>	<p><b>Java Fundamentals</b></p> <ol style="list-style-type: none"> <li>1. Write Java program to print “NIELIT, An Autonomous body of MeitY, GOP”.</li> <li>2. Write java program to print “Welcome to the year XX”. The XX is the year which should be accepted by the user.</li> <li>3. Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers.</li> <li>4. How to write comments in Java?</li> <li>5. Write a Java program to print the area and perimeter of a circle</li> <li>6. Write a Java program to swap two variables.</li> <li>7. Write a Java program to print the sum of the digits of three digits given number. Don’t use looping statement.</li> </ol>																		
<p><b><u>Assignment 2.</u></b></p>	<p><b>Control and Looping Statements</b></p> <ol style="list-style-type: none"> <li>1. Write a Java program to compare two numbers and show the result.</li> <li>2. Write a Java program that takes a number from the user and generates an integer between 1 and 7 and displays the name of the weekday.</li> <li>3. Write a Java program to print the odd numbers from 1 to 99. Prints one number per line.</li> <li>4. Write a Java program to print numbers between 1 to 100 which are divisible by 3, 5 and by both.</li> <li>5. Write a program to display the following output</li> </ol> <table data-bbox="478 1164 1069 1366"> <thead> <tr> <th>Pattern1</th> <th>Pattern2</th> <th>Pattern3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>*****</td> </tr> <tr> <td>22</td> <td>12</td> <td>****</td> </tr> <tr> <td>333</td> <td>345</td> <td>***</td> </tr> <tr> <td>4444</td> <td>6789</td> <td>**</td> </tr> <tr> <td>55555</td> <td></td> <td>*</td> </tr> </tbody> </table>	Pattern1	Pattern2	Pattern3	1	0	*****	22	12	****	333	345	***	4444	6789	**	55555		*
Pattern1	Pattern2	Pattern3																	
1	0	*****																	
22	12	****																	
333	345	***																	
4444	6789	**																	
55555		*																	
<p><b><u>Assignment 3.</u></b></p>	<p>OOP’s concept</p> <ol style="list-style-type: none"> <li>1. What is the use of super keyword? Explain with example.</li> <li>2. How multiple Inheritance can be achieved in Java? Explain with Java.</li> <li>3. Write a program to create parameterized constructor.</li> <li>4. Write a class Time with three data members to store hr, min, seconds. Create two constructors and apply checks to set valid time. Also create display function which displays all data members.</li> <li>5. Create an Abstract class Student that contains a method takeExam(), implement the method in the child classes PhdStudent and GradStudent in which PdhStudent takes exam by giving his final defense presentation while the graduate student gives a written paper.</li> </ol>																		
<p><b><u>Assignment 4.</u></b></p>	<p><b>Packages</b></p> <ol style="list-style-type: none"> <li>1. Write a Java Program using the concept of Packages.</li> <li>2. Write a Java program to perform employee payroll processing using packages. In the java file, Emp.java creates a package employee and creates a class Emp. Declare the variables name, empid, category, bpay, hra, da, npay, pf, grosspay, incometax, and allowance. Calculate the values in methods. Create another java file Emppay.java. Create an object e to call the methods to perform and print values.</li> </ol>																		

<p><b><u>Assignment 5.</u></b></p>	<p><b>Arrays and String</b></p> <ol style="list-style-type: none"> <li>1. Write a Java program to find the maximum and minimum value of an integer array.</li> <li>2. Write a java programme to copy all elements of one array into another array.</li> <li>3. Write a Java program to convert a given string into lowercase.</li> <li>4. Write a Java program to concatenate a given string to the end of another string.</li> <li>5. Write a Java program to get the index of all the characters of the alphabet.</li> </ol>
<p><b><u>Assignment 6.</u></b></p>	<p><b>Exception Handling</b></p> <ol style="list-style-type: none"> <li>1. Write a program that takes number from user and if the user provides alphabets as input, then handle this exception using try catch.</li> <li>2. Write a java program using multiple catch blocks. Create a class CatchExercise inside the try block declare an array a[] and initialize with value a[5] =30/5; . In each catch block show Arithmetic exception and ArrayIndexOutOfBoundsException.</li> <li>3. Write the Java program that uses the concept of finally block.</li> <li>4. Write the Java program that uses Java throw.</li> <li>5. Write the Java program that uses Java throws keyword.</li> </ol>
<p><b><u>Assignment 7.</u></b></p>	<p><b>Multithreading</b></p> <ol style="list-style-type: none"> <li>1. Can we override the Start Method? Explain with example.</li> <li>2. How can we start Custom Thread From the Overriding method? Explain with example.</li> </ol>
<p><b><u>Assignment 8.</u></b></p>	<p><b>Input/ Output Operations</b></p> <ol style="list-style-type: none"> <li>1. Write a Java program to get specific files by extensions from a specified folder.</li> <li>2. Write a Java program to check if a file or directory has read and write permission.</li> <li>3. Write a Java program to read a file content line by line.</li> <li>4. Write a Java program to get a list of all file/directory names from the given path.</li> </ol>
<p><b><u>Assignment 9.</u></b></p>	<p><b>Event Handling and Swing</b></p> <ol style="list-style-type: none"> <li>1. How events are handled in Java? Explain with example.</li> <li>2. Write a program using Swing as picture below, Each time the count button is clicked, the counter value shall increase by 1.</li> </ol> 
<p><b><u>Assignment 10.</u></b></p>	<p><b>Introduction to SQL, Database connectivity</b></p> <ol style="list-style-type: none"> <li>1. <b>Consider the following relations:</b> <ul style="list-style-type: none"> <li>S (sid, sname, status, city)</li> <li>SP (sid, pid, qty)</li> <li>P (pid, pname, color, weight, city)</li> </ul> </li> </ol> <p><b>Give an expression in SQL for each of queries below:</b></p> <ul style="list-style-type: none"> <li>• Get supplier names for supplier who supply at least one red part.</li> </ul>

	<ul style="list-style-type: none"><li>• Get supplier name for supplier who do not supply part p2.</li></ul> <p>2. <b>Consider the following relations:</b> <b>Employee</b> (emp_id, name) <b>Project</b> (project_id, project_name, chief_architect) <b>Assigned_to</b> (project_id, emp_id) <b>Give an expression in SQL for each of queries below:</b></p> <ul style="list-style-type: none"><li>• Print the details of the employee working on project 'AAA'</li><li>• Print the employee number of employee who work on all projects</li><li>• Print the emp_id of employee other than employee 39 who work on at least on project that employee 39 works on.</li></ul>
--	---