

**Course Title: Java Programming I**  
**Course No: CSIT.322**  
**Nature of the Course: Theory + Lab**  
**Year: Third, Semester: Sixth**  
**Level: B. Sc. CSIT**

**Credit: 3**  
**Number of period per week: 3+3**  
**Total hours: 45+45**

### 1. Course Introduction

This course introduces the fundamental programming concepts and techniques in Java. All elements of object-oriented programming are introduced. Topics covered include control structures, classes and objects, dynamic memory allocation, Inheritance and Polymorphism, File Handling, Multithreading, Exception Handling, and Generic Programming.

### 2. Objectives

Upon completion of this course students should:

- Understand the basic concepts and principles of object oriented programming.
- Be able to design, write and test a Java program to implement a working solution to a given problem specification.
- Be able to deal with exceptions effectively and write multithreaded programs

### 3. Specific Objectives and Contents

Specific Objectives	Contents
<ul style="list-style-type: none"><li>• Understand importance of java technology</li><li>• Setup java environment and get ready for coding</li><li>• Compile and Execute java programs</li><li>• Read inputs and Display Outputs</li></ul>	<b>Unit I: Java Programming Basics (4 Hrs)</b> <ul style="list-style-type: none"><li>1.1. History of java, Characteristics of java, Architecture of java</li><li>1.2. PATH and CLASSPATH Variables, Structure of Java Programs, Compiling &amp; Running Java Programs</li><li>1.3. Review of Data Types, Comments, Operators, Variables, Converting between Data Types (Type Casting), Strings, Arrays, Constants</li><li>1.4. Command Line Arguments, StringBuffer Class, Reading from Keyboard using Scanner Class, Using Math Class</li></ul>
<ul style="list-style-type: none"><li>• Use decision statements in programs</li><li>• Demonstrate looping statements and program them</li><li>• Apply jump statements in programs</li></ul>	<b>Unit II: Control Flow (4 Hrs)</b> <ul style="list-style-type: none"><li>2.1. Selection Statements: if statements, if....else statements, else if ladders, switch statements</li><li>2.2. Looping: While Loop, Do While Loop, For Loop, Enhanced For Loop</li><li>2.3. Jump Statements: Break Statement, Continue Statement, Return Statement</li></ul>
Understand class and objects and	<b>Unit III: Class and Objects (6 Hrs)</b>

develop programs around it. Use access Specifiers properly to class members Exemplify static data members and methods Understand constructors and use it in programs Pass arguments and return values from methods	3.1. Creating Classes, Defining member variables and methods, Creating Reference Variables, Creating Objects, Using member variables and methods 3.2. Access Specifiers: Public, Protected, Default, and Private 3.3. Static and Non-static members, Constructors, This Keyword, Garbage Collection, Inner Classes, Local Classes 3.4. Passing Parameters, Arrays, Objects to Methods and Constructors, Returning Values, Arrays, Objects from Methods and Constructors
<ul style="list-style-type: none"> <li>• Write polymorphic programs using overloading and overriding</li> <li>• Understand importance of inheritance and use it in writing programs</li> <li>• Explain concepts of containership and abstract classes</li> </ul>	<b>Unit IV: Inheritance and Polymorphism (6 Hrs)</b> 4.1. Method Overloading, Constructor Overloading, Creating Subclass, Different Types of Inheritance 4.2. Method Overriding, Dynamic Method Dispatch, Using Constructors and Inheritance, Super Keyword 4.3. Access Specifiers and Inheritance, Final Methods, Final Classes 4.4. Has-a Relationship (Containership), Object Class, Abstract Classes
<ul style="list-style-type: none"> <li>• Understand interfaces and use it in programs</li> <li>• Differentiate between interfaces and abstract classes.</li> <li>• Demonstrate packages by creating and using it.</li> </ul>	<b>Unit V: Interfaces and Packages(4 Hrs)</b> 5.1. Defining Interfaces, Interfaces vs. Classes, Extending Interfaces, Implementing Interfaces, Multiple Inheritance by using interfaces, Abstract Classes vs. Interfaces. 5.2. Importance of Packages, Using Packages, Creating Packages
<ul style="list-style-type: none"> <li>• Read inputs from files and store outputs in files.</li> <li>• Understand and use byte stream classes and character stream classes</li> <li>• Use random access and tokenizer in files</li> </ul>	<b>Unit VI: File and IO Handling (5 Hrs)</b> 6.1. Concept of IO Streams, File Class, InputStream and OutputStream Class, FileInputStream and FileOutputStream Class, BufferedInputStream and BufferedOutputStream Class 6.2. Reader and Writer Classes, FileReader and FileWriter Class, InputStreamReader and OutputStreamWriter Class, BufferedReader and BufferedWriter Class, 6.3. Random File Access, StreamTokenizer Class, Using PrintWriter Class, Using Scanner Class
<ul style="list-style-type: none"> <li>• Understand exceptions and its categories</li> <li>• Hand exceptional conditions in programs by using different keywords</li> <li>• Define own exception classes and use them in exception handling</li> </ul>	<b>Unit VII: Exception Handling (5 Hrs)</b> 7.1. Concept of Exception and Exception Handling, Categories of Exceptions, Hierarchy of Exception Classes 7.2. Using Try....Catch, Multiple Catch Blocks, Finally Keyword 7.3. Using Throws and Throw Keywords, Nested Try....Catch, Creating Exception Classes
<ul style="list-style-type: none"> <li>• Explain importance of</li> </ul>	<b>Unit VIII: Multithreading (5 Hrs)</b>

multithreaded programs • Use Runnable interface and Thread class in creating threads • Understand thread life cycle and manage multithreaded programs by using different methods.	8.1. Concept of Thread and Multithreading, Main Thread, Naming a Thread, Pausing a Thread, Thread Life Cycle 8.2. Multithreading by Using Runnable Interface, Multithreading by using Thread Class, Creating multiple threads, Joining Threads, setting Thread Priority, Stopping Threads 8.3. Thread Synchronization, Communication between Threads, Suspending and Resuming Threads
• Understand generics and write generic java programs • Understand collection framework and use collection classes	<b>Unit IX: Generics and Collection Classes(5 Hrs)</b> 9.1. Concept of Generics, Generic Methods, Bounded Type Parameters, Generic Classes 9.2. Collections and Collection Framework, Collection Classes ( Stack, Linked List, Hash Table), Iterator, Comparator

## Evaluation System

Undergraduate Programs							
External Evaluation	Marks	Internal Evaluation	Weight age	Marks	Practical	Weight age	Mark
End semester examination	60	Assignments	20%	20	Practical Report copy	25%	20
(Details are given in the separate table at the end)		Quizzes	10%		Viva	25%	
		Attendance	20%		Practical Exam	50%	
		Internal Exams	50%				
Total External	60	Total Internal	100%	20		100%	20
Full Marks 60+20+20 = 100							

### External evaluation

#### 1. End semester examination:

It is a written examination at the end of the semester. The questions will be asked covering all the units of the course. The question model, full marks, time and others will be as per the following grid.

#### 2. External Practical Evaluation:

After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the above mentioned evaluation. There will be an internal examiner to assist the external examiner.

Three hours time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Full Marks: 100, Pass Marks: 45, Time: 3 Hrs

Nature of question	Total questions to be asked	Total questions to be answered	Total marks	Weightage
Group A: multiple choice*	20	20	$20 \times 1 = 20$	60%
Group B: Short answer type questions	7	6	$6 \times 8 = 48$	60%
Group C: Long answer type questions	3	2	$2 \times 16 = 32$	60%
			100	100%

Each student must secure at least 50% marks in internal evaluation in order to appear in the end semester examination. Failed student will not be eligible to appear in the end semester examinations.

### Internal evaluation

**Assignment:** Each student must submit the assignment individually. The stipulated time for submission of the assignment will be seriously taken.

**Quizzes:** Unannounced and announced quizzes/tests will be taken by the respective subject teachers. Such quizzes/tests will be conducted twice per semester. The students will be evaluated accordingly.

**Attendance in class:** Students should regularly attend and participate in class discussion. Eighty percent class attendance is mandatory for the students to enable them to appear in the end semester examination. Below 80% attendance in the class will signify NOT QUALIFIED (NQ) to attend the end semester examination.

**Presentation:** Students will be divided into groups and each group will be provided with a topic for presentation. It will be evaluated individually as well as group-wise. Individual students have to make presentations on the given topics.

**Mid-term examination:** It is a written examination and the questions will be asked covering all the topics in the session of the course.

**Discussion and participation:** Students will be evaluated on the basis of their active participation in the classroom discussions.

**Instructional Techniques:** All topics are discussed with emphasis on real-world application. List of instructional techniques is as follows:

- Lecture and Discussion
- Group work and Individual work
- Assignments

- Presentation by Students
- Quizzes
- Guest Lecture

Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during that period. If a student fails to attend a formal exam/quiz/test, there won't be any provision for re-exam. Unless and until the student clears one semester he/she will not be allowed to study in the following semesters.

### **Laboratory Work**

Student should write programs and prepare lab sheet for most of the units in the syllabus. They should practice design and implementation of java programs that demonstrates different concepts discussed in class. However, nature of programming can be decided by the instructor. The lab work should be practiced for minimum of 3 lab hours per week.

### **Prescribed Text**

- Cay S. Horstmann, Core Java Volume I--Fundamentals Ninth Edition, Prentice Hall, 2012

### **References**

- Hebert Schildt Java: The Complete Reference, McGraw-Hill Education, Ninth Edition, 2014
- Steven Holzner, Java 7 Programming, Black Book, Dreamtech Press, 2013