

# J2SE Syllabus

---

Core Java (till JDK 1.7)

sterling

institute

STERLING INSTITUTE

110, Fortune Aura Business Park, near Apple Hospital, Bhawarkua, Indore-452014 (M. P.)

+91 9109913455

[www.sterlinginstitute.in](http://www.sterlinginstitute.in)

# Sterling on Air

**“Programming is in the air”**

We believe in providing quality not only certification

We provide Corporate Trainings in Industry and Colleges:

- ✓ Java
- ✓ C & C++
- ✓ Business Intelligence
- ✓ Data Analytics
- ✓ Python
- ✓ Android / Kotlin
- ✓ Flutter
- ✓ UI/ UX
- ✓ Full Stack Java
- ✓ Full Stack Python
- ✓ Full Stack Front End
- ✓ Django | Flask
- ✓ Data Science
- ✓ Oracle
- ✓ MERN Stack
- ✓ DBMS with SQL & PL/ SQL
- ✓ Industrial Trainings
- ✓ Placement Oriented Trainings
- ✓ Corporate Trainings
- ✓ Student Development Program
- ✓ Faculty Development Program



# Index

S. No.	Chapters
1.	Introduction
2.	Logical Programming
3.	String Handling
4.	Variables & Operators
5.	Command Line Arguments & Math Handling
6.	Array
7.	Methods
8.	Import & Package Statement
9.	Object Oriented Programming through Java
10.	Date, Time & Currency
11.	Exception Handling
12.	Collection Framework
13.	File IO
14.	NIO
15.	Multi-Threading
16.	Concurrency Utilities
17.	Regular Expression
18.	Reflection API
19.	JVM Architecture
20.	Garbage Collection
21.	Event Handling & AWT
22.	Swing / JFX
23.	Java Commands
24.	Compiler & JVM Warnings
25.	Project: Desktop Application

<b>CHAPTER: 1 Introduction</b>	
<b>1. Computer</b>	
a. Basic Components of Computer	
b. Types of Computers	
c. Software & its Types	
d. Basic Foundation of Computer	
e. Generations of Computers	
f. MS Dos (Microsoft Disk Operating System) Command	
<b>2. Introduction of Programming Languages</b>	
a. What are Languages?	
b. What is Machine?	
c. Types of Translators	
i. Assembler	
ii. Compiler	
iii. Interpreter	
d. Types of Programming Languages	
e. Programming Languages Vs Scripting Languages	
f. Natural Languages v/s Programming Languages	
<b>3. Introduction to Java</b>	
a. Genesis of Java	
b. Java Versions	
c. Features of Java	
d. Scope of Java	
e. Java Platform	
f. Configuration	
g. Class	
h. Java Keywords	
i. Universal Standard	
j. Class Name Rules	
k. Executable Class	
l. First Java Program	
m. Compilation Process	
n. Compiler Responsibilities in Java	
o. Print Statement	
p. Main Method Manipulations	
q. Byte Code Generation	
r. Key Points Related to Byte Code	
s. Operating System Reserved Words	
t. Java 9 Class Execution Feature	
u. Platform Independent or WORA	
v. Portability	
w. Architecture Neutral	
<b>4. Java Editions</b>	
a. J2SE	
b. J2EE	
c. J2ME	
<b>CHAPTER : 2 Logical Programming</b>	
<b>1. Decision Making Statements</b>	
a. if	
b. if else	
c. Ladder if else	
d. Nested if else	
<b>2. Iterational Statements Or Loops</b>	
a. while	
b. for	
c. Nested for loop	
d. Nested while loop	
e. do while loop	
f. Nested do while loop	
g. Labelling of loops	
h. Labelling of Statements	
i. for-each loop	
j. Nested for-each loop	
k. Pattern Programming	
l. Manipulations of Control statements	
m. Manipulations of if else	

n. Manipulations of while loop
o. Manipulations of for loop
p. Manipulations of do while loop
q. Manipulations of for-each loop
<b>3. Branching Statements</b>
a. break statement
b. continue statement
c. Return statement
d. goto statement
<b>4. Multi-Way Branching Statement</b>
a. switch-case statement
<b>CHAPTER : 3 String Handling</b>
a. In-built Class Name Vs Custom Class Name
b. String Class
c. String Class Methods & Constructors
d. CharSequence
e. StringBuffer Class
f. StringBuffer Class Methods
g. StringBuilder Class (JDK 1.5)
h. StringBuilder Class Methods
i. Ways to define String in Java
j. String Memory Areas
k. String Literal Vs String Object
l. '==' operator Vs equals() method
m. String Vs StringBuffer Vs StringBuilder
n. StringTokenizer Class
o. StringTokenizer Class Methods & Constructors
p. StringJoiner Class (JDK 1.8)
q. String Joiner Class Methods & Constructors
<b>CHAPTER: 4 Variables &amp; Operators</b>
<b>1. Variables</b>
a. Variable Declaration
b. Dynamic Variable Initialization

c. Scope & Lifetime of Variable
d. Data types
i. Primitive Data Type
ii. Reference Data Type
iii. Custom Data Type
<b>2. Type Conversion</b>
a. Implicit type conversion
b. Explicit Type Conversion
c. Automatic Type Promotion
d. Type Promotion Rules
e. Primitive To String Type Conversion
f. String To Primitive Type Conversion
<b>3. Literals</b>
a. Integer Literals
b. Floating-Point Literals
c. strictfp keyword
d. Character Literals
e. String Literals
f. Boolean Literals
g. Null Literal
<b>4. Representation and Conversion of Number System</b>
a. Binary to Decimal Conversion
b. Decimal to Binary Conversion
c. Binary to Octal Conversion
d. Octal to Binary Conversion
e. Binary to Hexadecimal Conversion
f. Hexadecimal to Binary Conversion
<b>5. Operators</b>
a. Unary Operator
i. Increment Operator
ii. Decrement Operator
iii. Complement Operator
b. Binary Operator
i. Arithmetic Operator

ii. Relational Operator
iii. Logical Operator
iv. Bitwise Operator
v. Assignment Operator
vi. Compound Operator
c. Ternary Operator
d. Miscellaneous Operator
i. Concatenation Operator
ii. Dot Operator
iii. instanceof Operator
e. Operator Precedence & Associativity
f. Advantages of Operators
g. Limitations of Operators
<b>Chapter 5: Command Line Arguments &amp; Math Handling</b>
<b>1. Command Line Arguments</b>
a. Command Line Menu
<b>2. Math Class</b>
a. General Methods
b. Trigonometry Methods
c. Rounding Methods
<b>Chapter 6: Array</b>
<b>1. One-Dimensional Array</b>
a. Introduction of Array
b. Array Declaration
c. Types Array Declaration
d. Array Initialization
e. Types of Array Initialization
f. Accessing Array Elements
g. length Attribute
h. One-dimensional Array in Memory
i. Merging Two Arrays
j. Traversing of Array
k. Primitive Elements Array

l. Non-Primitive Elements Array
m. Char Array and its feature
n. Array Copy
<b>2. Searching &amp; Sorting Algorithms</b>
a. Linear Searching Algorithm
b. Binary Searching Algorithm
c. Insertion Sorting Algorithm
d. Bubble Sorting Algorithm
e. Quick Sorting Algorithm
f. Merge Sorting Algorithm
<b>3. Multidimensional Arrays</b>
a. Multi-Dimensional Array Declaration
b. Types of Declaration
c. Row-Major Implantation
d. Column-Major Implementation
e. Initializing Multi-Dimensional Array
f. Accessing Multi-Dimensional Array
<b>4. Jagged Arrays</b>
<b>Chapter 7: Methods</b>
<b>1. Introduction to Method</b>
a. Methods in Programming Languages
i. Inbuilt Method
ii. Custom Method
b. Methods Definition
c. Method Signature
d. Parts of Method
e. Method Naming Convension
f. Method Calling
g. Non-Static & static Method Calling
h. Method Calling in Memory
i. Passing Array in Method
j. Advantages of Method
k. Patterns of Method
<b>2. Recursion</b>

a. Types of Recursion
<b>Chapter 8: Import &amp; Package Statement</b>
<b>1. Import Statement</b>
a. Types of import Statement
b. Import Vs Include
c. Importing sub packages
d. Static Import (JDK 1.5)
<b>2. Package Statement</b>
a. Introduction to package
b. Types of packages
i. In-built package
ii. Custom Package
c. Defining a Package
d. Package Compilation
<b>3. JAR</b>
a. Creating a JAR file
b. Manifest File
c. Common Pitfalls in Packages
<b>Chapter 9: Object Oriented Programming through Java</b>
<b>1. Introduction</b>
a. POP V/s OOP
b. Attribute
c. Class Diagram
<b>2. Setter Getter or Java Bean</b>
a. Rules for setter getter
b. this keyword for variables
c. Member Attributes & Member Methods
<b>3. Introduction to UML Diagrams</b>
<b>4. Constructors</b>
a. Types of Constructors
b. Default Constructor
c. Constructor Overloading
d. Destructor in C++

<b>5. Type of Variables in Java</b>
a. Instance Variable
b. Class Variable
c. Local Variable
d. var Keyword (JDK 10)
e. Memory Allocation of Variables
f. Object Instantiation
g. Object Characteristics
h. new Keyword
<b>6._INITIALIZER Blocks</b>
a. Static Block
b. Instance Block
c. Instance Initialization Block With Parent
d. Order of Execution
<b>7. Encapsulation</b>
a. Benefits of Encapsulation
b. Achieve Encapsulation
<b>8. Inheritance</b>
a. Types of Inheritance
b. extends keyword
c. Benefits of Inheritance
d. Limitations of Inheritance
e. Generalization Vs Specialization
f. Type Casting of Reference Datatype
<b>9. Object Class</b>
<b>10. Polymorphism</b>
a. Parent can keep child's Reference
b. Types of Polymorphism
i. Dynamic Polymorphism
ii. Static Polymorphism
<b>11. Dynamic Polymorphism</b>
a. Method Overriding
b. Rules for Method Overriding
c. Co-Variant Return Type (JDK 1.5)
d. @Override Annotation (JDK 1.5)

e. Access Level Modifiers
f. Runtime Resolution
g. final Keyword
h. static Vs final
<b>12. abstract keyword</b>
a. Abstract Class
b. Abstract Method
c. Benefits of abstract keyword
d. Limitation of abstract keyword
<b>13. Uses of Polymorphism</b>
a. Polymorphism in Array
b. Polymorphism in Method Argument
c. Polymorphism in method return type
<b>14. Static Polymorphism</b>
a. Method Overloading
i. Rules for Method Overloading
ii. Primitive Parameter Manipulations
iii. Reference Parameter Manipulations
b. Var-Args (JDK 1.5)
i. Rules for Var-Args
c. Method Hiding
i. Rules for Method Hiding
ii. Method Overriding Vs Method Hiding
d. Multiple ways to create objects
e. Singleton Design Pattern
f. Factory Design Pattern
<b>15. Interface</b>
a. <b>Interface (JDK 1.0 - JDK 1.7)</b>
b. Introduction
c. Rules for Interface
d. Interface Attribute
e. Interface Method
f. Interface Inheritance
g. Multiple Inheritance in Interface
h. Interface Blocks

i. implements Keyword
j. Interface UML
k. Interface Naming Conventions
l. Bridge Design Pattern
m. Types of Interface
n. Marker Interface
o. @Functional Interface (JDK 1.8)
p. instanceof keyword
q. <b>Interface JDK 1.8</b>
r. Interface static Methods
s. Interface default Methods
t. <b>Interface JDK 1.9</b>
u. Interface private Methods
v. Interface private static Methods
w. Sealed classes and Interfaces (JDK 17)
<b>16. Comments</b>
a. // Single Line Comment
b. /* Multi-Line Comment */
c. /** Documentation Comment */
d. javadoc Command
<b>17. Data Abstraction</b>
a. Standards of Abstraction
b. Data Hiding
<b>18. Constructor Invocation</b>
a. this() constructor call
b. super() constructor call
c. this Method call
d. super Method Call
<b>19. Inner Class or Nested Class</b>
a. Need of Inner class
b. Types of Inner Class
i. Non-Static Inner Class
• Member Inner Class
• Anonymous Inner Class



• Local Inner Class
ii. Static Inner Class
<b>20. Enum (JDK 1.5)</b>
a. Enumeration Fundamentals
b. enum Keyword
c. Enum Declaration.
d. Enum Usage
e. Enum Examples
f. Enum Class
g. value() and valueOf()
<b>21. Annotations (JDK 1.5)</b>
a. Annotation Fundamentals
b. XML Vs Annotations
c. Declaration of Annotations
d. Annotation Calling
e. AnnotatedElement Interface
f. Annotation Members
g. Categories of Annotations by its Members
h. Types of Annotation
i. Standard Annotation
a) General Purpose Annotations
b) Meta-Annotations
ii. Custom Annotation
<b>22. Object Cloning</b>
i. clone() method
ii. Cloneable Interface
iii. Shallow Cloning
iv. Deep Cloning
v. Copy Constructor
<b>Chapter: 10 Date, Time &amp; Currency</b>
<b>1. Date Handling</b>
a. Date Class & its Methods
b. Date Class Constructors
c. Unix Time

d. DateFormat Class
e. SimpleDateFormat Class
f. Calendar class
g. GregorianCalendar Class
<b>2. Time Handling</b>
a. TimeZone Class
b. SimpleTimeZone Class
c. Locale Class
d. LocalDate Class (JDK 1.8)
e. LocalTime Class (JDK 1.8)
f. LocalDateTime API (JDK 1.8)
g. ZonedDateTime Class
h. DateTimeFormatter Class
i. Period Class
j. Duration Class
k. Month Class
l. Formatting Date & Time
<b>3. Currency Handling</b>
a. Currency Class
b. Formatter Class
c. Specifying Precision
d. Format Flags
<b>Chapter : 11 Exception Handling</b>
<b>1. Introduction to Exception</b>
a. Exception Fundamentals
b. Exception-Handling Fundamentals
c. Uncaught Exception
d. Exception Types
e. Exception Description
f. Java's Built-in Exceptions
<b>2. Exception Handling Mechanism</b>
a. try block
b. catch block
c. finally block

d. Resource releasing code
e. try with Multiple catch
f. try with finally
g. Nested try catch finally
h. System.exit(0);
i. Handling in Primitive Datatype
j. printStackTrace();
k. getMessage();
<b>3. Exception Hierarchy</b>
a. Throwable Class & its Constructors
b. Error Class & its Methods
c. Exception Class & its Methods
d. Error Vs Exception
e. Checked Exception Vs Unchecked Exceptions
f. Generic catch block
g. Order of catch blocks
<b>4. Custom Exception</b>
a. Creating own Exception Classes
b. throw keyword
c. Creating our own messages
<b>5. Exception Propagation</b>
a. throws keyword
b. Propagation Order
c. Chained Exception
<b>6. Three Features introduces in JDK 1.7</b>
a. try-with-resources (JDK 1.7)
b. multi-catch
c. final rethrow or more precise rethrow
<b>7. Experiments with Methods and Constructors</b>
a. Return a value from try, catch and finally
b. Throwing Exception from Constructors
<b>8. Errors</b>
a. Error Fundamentals
b. Types of Errors
c. Error Hierarchy

<b>Chapter 12: Collection Framework</b>
<b>1. Fundamentals of Collection Framework</b>
a. Limitations of using Manual Data Structures
b. Need of Collection Framework
c. Heterogeneous Elements in Array
d. Array V/s Collection
e. Collection Hierarchy
f. Map Hierarchy
<b>2. Wrapper Classes</b>
a. Memory Allocations
b. Numeric Type Wrapper
c. Boolean Type Wrapper
d. Character Type Wrapper
<b>3. Auto-Boxing &amp; Auto-Unboxing (JDK 1.5)</b>
<b>4. Collection (JDK 1.2)</b>
a. Collection Interface & its Methods
b. Iterable Interface
c. Collection vs. Map
<b>5. List Interface</b>
a. ArrayList Class
b. Internal Working of ArrayList
c. RandomAccess Interface
d. LinkedList Class & its Methods
e. Internal Working of LinkedList
f. Vector Class & its Methods
g. Internal Working of Vector
h. ArrayList vs. LinkedList
i. ArrayList vs. Vector
j. Stack Class & its Methods
k. Internal Working of Stack
<b>6. Set Interface (Part I)</b>
a. HashSet Class & its Methods
b. Internal Working of HashSet
c. LinkedHashSet Class (JDK 1.4)

d. Internal Working of LinkedHashSet
e. Hashset vs. LinkedHashSet
<b>7. Cursors</b>
a. Iterator Interface
b. Enumeration Interface
c. ListIterator Interface
d. For-Each
e. Spliterator Interface
<b>8. Object Class Methods</b>
a. hashCode()
b. equals()
c. hashCode() and equals() Contract
d. getClass() method
e. toString() method
<b>9. Sorting</b>
a. Comparable Interface
b. Comparator Interface
c. Comparable vs. Comparator
<b>10. Set Interface (Part II)</b>
a. SortedSet Interface & its Methods
b. NavigableSet Interface
c. TreeSet Class & its Methods
d. Internal Working of TreeSet
e. Balanced Binary Tree
f. Traversal Techniques of Binary Tree
g. StringBuffer objects in TreeSet
h. Custom Object Sorting in TreeSet
<b>11. Queue Interface (JDK 1.5)</b>
a. PriorityQueue Class
b. Internal Implementation of PriorityQueue
c. Deque Interface
d. BlockingQueue Interface
e. PriorityBlockingQueue Class
f. LinkedBlockingQueue Class
g. ArrayDeque Class

<b>12. Utility Classes</b>
a. BitSet Class
b. Random Class
c. Arrays Class
d. Collections Class
<b>13. Map (JDK 1.2)</b>
a. Map Interface & its Methods
b. HashMap Class
c. Internal Working of HashMap
d. LinkedHashMap Class
e. Internal Working of LinkedHashMap
f. HashMap vs. LinkedHashMap
g. WeakHashMap Class
h. Internal Working of WeakHashMap
i. IdentityHashMap Class
j. Internal Working of IdentityHashMap
k. SortedMap Interface & its Methods
l. NavigableMap Interface & its Methods
m. TreeMap Class
n. Red-Black Tree
o. Internal Implementation of TreeMap
p. Hashtable Class
q. Internal Working of Hashtable
r. Dictionary Class
s. Properties Class
t. store() and load()
<b>14. Generics</b>
a. Introduction to Generics
b. Simple Generic Example
c. Syntax and Usage of Generics
d. Generic Class with Two Type Parameters
e. Bounded Type Parameters
f. Generics with Collections Framework
g. Generic Methods
h. Using Wildcards Arguments

i. Restrictions on Generics
j. Generics Class Hierarchies
k. Generic Interfaces
l. Type Erasure
m. Type Interface with Generics
n. Ambiguity Errors
o. Generic Restrictions
p. Restrictions on Static Members
q. Generic Array Restrictions
r. Generics with Enums and Exceptions
<b>Chapter 13: File IO</b>
<b>1. Fundamentals of File IO</b>
a. Types of Memories
b. Types of Data
c. Types of Streams
<b>2. File Class</b>
a. Representation of Files
b. Directories
c. Drives
d. File Class Methods
<b>3. Character Based Stream</b>
a. Reader Class
b. FileReader Class
c. BufferedReader Class
d. PushbackReader Class
e. CharArrayReader Class
f. Writer Class
g. FileWriter Class
h. BufferedWriter Class
i. CharArrayWriter Class
j. PrintWriter Class
k. AutoCloseable, Closeable and Flushable
l. try-with-resources (JDK 1. 7)
<b>4. Byte Based Stream</b>

a. InputStream Class
b. FileInputStream Class
c. BufferedInputStream Class
d. ByteArrayInputStream Class
e. OutputStream Class
f. FileOutputStream Class
g. BufferedOutputStream Class
h. ByteArrayOutputStream
i. PrintStream Class
j. InputStreamReader Class
<b>5. Serialization</b>
a. ObjectOutputStream Class
b. transient keyword
c. NotSerializableException Class
d. Deserialization
e. ObjectInputStream Class
f. Externalization
g. Externalizable Interface
h. DataOutputStream Class
i. DataInputStream Class
j. RandomAccessFile Class
k. Scanner Class
l. Console Class
m. System.out
n. System.in
o. System.err
<b>Chapter 14: NIO</b>
<b>1. Fundamentals of NIO</b>
<b>2. NIO Classes</b>
<b>3. Buffers</b>
<b>4. Channels</b>
<b>5. Charsets and Selectors</b>
a. Path Interface (JDK 1.7)
b. Files Class (JDK 1.7)

c. Paths Class (JDK 1.7)
d. FileSystem, FileSystems, FileStore (JDK 1.7)
<b>6. Channel Based IO</b>
<b>7. Stream Based IO</b>
<b>Chapter 15: Multi-Threading</b>
<b>1. Fundamentals of Multithreading</b>
a. Introduction to Multithreading
b. Application of Multithreading
c. Examples of Multithreading
d. Single Threaded Program
<b>2. Multitasking</b>
a. Types of Multitasking
b. Process-Based Multitasking
c. Thread Based Multitasking
d. Process
e. Thread
<b>3. Thread Definition</b>
a. Types of Thread Definition
b. Extends Thread Class
c. Implements Runnable Interface
d. start() Method Responsibilities
e. Thread Registration
f. Resources Allocation
g. run() Method
h. Thread Class Methods
i. Thread Class Constant
j. Thread Priority
k. ThreadGroup Class
l. Runnable Interface
m. Prevent Thread Execution
Inter-Thread Communication
n. Object Class Methods
o. Thread Life Cycle
p. Thread State

q. Shutdown Hook
r. Exception in Thread
<b>4. Thread Models</b>
a. Native Thread Model
b. Green Thread Model
<b>5. Java Scheduling</b>
a. Scheduling
b. Thread Scheduler
c. Preemptive Scheduling
d. Non-Preemptive Scheduling
e. Deadlock
f. Starvation or Livelock
g. Time Sliced Scheduling
h. Non-Time Sliced Scheduling
i. Daemon Thread
j. Race Condition
k. Synchronized keyword
l. Synchronized Method & Block
m. Static Synchronization
n. Main method Manipulations
o. Monitor
p. Object Monitor
q. Class Monitor
r. Re-entrant Monitor
s. Runtime Class
<b>Chapter 16: Concurrency Utilities</b>
1. Locks
2. Atomic
3. Semaphore
4. CyclicBarrier
5. Exchanger
6. Phaser
7. Executor Framework
a. Types of Executor

b. Single Thread Executor
c. Fixed Thread Pool (n)
d. Cached Thread Pool
e. Scheduled Executor
8. Concurrent Collection
9. Atomic Operators
10. Fork/Join Framework
a. The Main Fork/ Join Classes
b. The Divide-and-Conquer Strategy
c. Impact of the Level of Parallelism
d. Executing a task Asynchronously
e. Cancelling a Task
f. Determine a Task's Completion Status
g. Restarting a Task
h. Concurrency Vs. Traditional Approach
<b>Chapter 17: Regular Expression</b>
<b>1. Regular Expression Processing</b>
a. Pattern
b. Matcher
c. Regular Expression Syntax
d. Demonstrating Pattern Matching
e. Two Pattern-Matching Options
f. Exploring Regular Expressions
<b>Chapter : 18 Reflection API</b>
a. Introduction to Reflection API
b. Ways to get Class class object
c. Accessing Class Information
d. Accessing Constructors
e. Accessing Fields
f. Accessing Methods
g. Accessing Annotations
h. Accessing Interfaces and Superclasses
i. Handling Arrays with Reflection
j. Accessing and Modifying Static Members

k. Reflection and Generics
l. Dynamic Proxies (Advanced)
m. Security and Performance Considerations
n. Implement javap tool
o. Call private method
<b>Chapter 19: JVM Architecture</b>
<b>1. Introduction to Virtual Machine</b>
a. Types of Virtual Machine
b. Importance of Virtual Machine
<b>2. Introduction to JVM</b>
a. Components of JVM
b. JVM Block Diagram
c. Class Loader Sub System
d. Type of Class Loaders
i. Bootstrap Class Loader
ii. Extension Class Loader
iii. Application Class Loader
iv. Custom Class Loader
e. Working of Class Loader Sub System
f. Delegation Hierarchy Algorithm
g. JVM Memory Areas
i. Method Memory Area
ii. Heap Memory Area
iii. Stack Memory Area
iv. PC Register
v. Native Method Stack
h. Execution Engine
i. Interpreter
j. JIT Compiler
k. Working of JIT Compiler
l. JVM Execution Flow
m. JIT Compiler and Performance Optimization
n. Class File Structure
o. Magic Number

p. JVM Tuning and Configuration
<b>Chapter 20: Garbage Collection</b>
a. Introduction to Garbage Collection
b. Working of Garbage Collection
c. Garbage Collection on different Memories
d. Types of Garbage Collectors
e. GC Phases: Mark, Sweep, and Compact
f. Young Generation Garbage Collection
g. Old Generation Garbage Collection
h. Finalization
i. Weak, Soft, and Phantom References
j. Garbage Collection Tuning and JVM Options
k. Common Garbage Collection Issues
l. Monitoring and Analyzing Garbage Collection
m. Garbage Collection in Java 8 and Beyond
<b>Chapter :21 Event Handling &amp; AWT</b>
<b>1. Event Handling</b>
a. Two Event Handling Mechanism
b. The Delegation Event Model
i. Events
ii. Event Sources
iii. Event Listeners
c. Event Classes
i. The ActionEvent Class
ii. The AdjustmentEvent Class
iii. The ComponentEvent Class
iv. The ContainerEvent Class
v. The FocusEvent Class
vi. The InputEvent Class
vii. The ItemEvent Class
d. The KeyEvent Class
i. The MouseEvent Class
ii. The MouseWheelEvent Class
iii. The TextEvent Class
iv. The WindowEvent Class
e. Sources of Events

f. Event Listener Interfaces
i. The ActionListener Interfaces
ii. The AdjustmentListener Interfaces
iii. The ComponentListener Interfaces
iv. The ContainerListener Interfaces
v. The FocusListener Interfaces
vi. The ItemListener Interfaces
vii. The KeyListener Interfaces
viii. The MouseListener Interfaces
ix. The MouseMotionListener Interfaces
x. The MouseWheelListener Interfaces
xi. The TextListener Interfaces
xii. The WindowFocusListener Interfaces
xiii. The WindowListener Interfaces
g. Using the Delegation Event Model
i. Handling Mouse Event
ii. Handling Keyboard Events
h. Adapter Classes
i. Inner Classes
i. Anonymous Inner Classes
<b>2. Introducing The AWT: Working with Windows, Graphics, Text</b>
a. AWT Classes
b. Window Fundamentals
i. Component
ii. Container
iii. Panel
iv. Window
v. Frame
vi. Canvas
c. Working with Frame Windows
i. Setting the Window's Dimensions
ii. Hiding and Showing a Window
iii. Setting a Window's Title
iv. Closing a Frame Window
d. Creating a Frame Window in an Applet
i. Handling Events in a Frame Window
e. Creating a Windowed Program



f. Displaying Information Within a Window	j. Handling Scroll Bars
g. Working with Graphics	k. Using a Text Field
i. Drawing Lines	l. Handling a TextField
ii. Drawing Rectangles	m. Using a TextArea
iii. Drawing Ellipses and Circles	n. Understanding Layout Managers
iv. Drawing Arcs	o. Flow Layout
v. Drawing Polygons	p. Border Layout
vi. Sizing Graphics	q. Using Insets
h. Working with Color	r. GridLayout
i. Color Methods	s. CardLayout
ii. Setting the Current Graphics Color	t. GridBagLayout
iii. A Color Demonstration Applet	u. Menu Bars and Menus
i. Setting the Paint Mode	v. Dialog Boxes
j. Working with Fonts	w. FileDialog
i. Determining the Available Fonts	x. Handling Events by Extending AWT Components
ii. Creating and Selecting a Font	y. Extending Button
iii. Obtaining Font Information	z. Extending Checkbox
k. Managing Text Output using FontMetrics	aa. Extending Check Box Group
i. Displaying Multiple Lines of Text	bb. Extending Choice
ii. Centering Text	cc. Extending List
iii. Multiline Text Alignment	dd. Extending Scrollbar
<b>3. Using AWT Controls, Layout Managers and Menus</b>	ee. A Word About Overriding Paint()
a. Control Fundamentals	<b>4. Images</b>
i. Adding and Removing Controls	a. File Formats
ii. Responding to Controls	b. Image Fundamentals: Creating, Loading and Displaying
iii. The HeadlessException	c. Creating an Image Object
b. Labels	d. Loading an Image
c. Using Buttons	e. Displaying an Image
i. Handling Buttons	f. Image Observer
d. Applying Check Boxes	g. Double Buffering
i. Handling Check Boxes	h. MediaTracker
e. CheckBoxGroup	i. Image Producer
f. Choice Controls	j. Double Buffering
i. Handling Choice Lists	k. MediaTracker
g. Using Lists	l. ImageProducer
h. Handling Lists	m. Memory Image Source
i. Managing Scroll Bars	



n. ImageConsumer
o. PixelGrabber
p. ImageFilter
q. CropImageFilter
r. RGBImageFilter
s. Cell Animation
t. Additional Imaging Classes

## Chapter 22: Swing / JFX

### 1. Introducing SWING

a. The Origins of Swing
b. Swing is Built on AWT
c. Two Key Swing Features
i. Swing Components are Lightweight
ii. Swing Supports a Pluggable Look and Feel
d. The MVC Connection
i. Components and Containers
ii. Components
iii. Containers
iv. The Top-Level Container Panes
e. The Swing Packages
f. A Simple Swing Application
g. Event Handling
h. Create a Swing Applet
i. Painting in Swing
i. Painting Fundamentals
ii. Compute the Paintable Area
iii. A Paint Example

### 2. Exploring Swing

a. JLabel and ImageIcon
b. JTextField
c. The Swing Buttons
i. JButton
ii. JToggleButton
iii. Check Boxes
iv. Radio Buttons
d. JTabbedPane

e. JList
f. JComboBox
g. Trees
h. JTable

## Chapter 23: Java Commands

## Chapter 24: Compiler & JVM Warnings

## Project: Desktop Application

## Our Placements:

We've provided a course module to the pass out engineering candidates which strongly helped them to crack machine test and technical interviews easily.

Our students are placed in



IMPETUS

Infosys®

Rakuten

TATA  
CONSULTANCY  
SERVICES

YASH<sup>SM</sup>  
Technologies  
*More than what you think.*

consultadd

InfoBeans  
CREATING WOW!

LTI

amdocs

sYstango

FUJITSU

zehntech

Microsoft

Optum

gslab

inn@eye

<epam>

Tech  
Mahindra

CREDIT SUISSE



bitwise

VISIONWAVES

NSE