

**UNIT I****Chapter 1 : Fundamentals of Programming****1-1 to 1-19**

Importance of Studying Programming Languages, History of Programming Languages, Impact of Programming Paradigms, Role of Programming Languages, Programming Environments. Impact of Machine Architectures : The operation of a computer, Virtual Computers and Binding Times.

**Programming paradigms** : Introduction to programming paradigms, Introduction to four main Programming paradigms-procedural, object oriented, functional and logic and rule based.

<b>1.1</b>	<b>Importance of Studying Programming Languages.....</b>	<b>1-1</b>
<b>1.2</b>	<b>History of Programming Languages.....</b>	<b>1-2</b>
<b>1.3</b>	<b>Impact of Programming Paradigms .....</b>	<b>1-4</b>
<b>1.4</b>	<b>Role of Programming Languages.....</b>	<b>1-4</b>
1.4.1	A Simple Program.....	1-4
<b>1.5</b>	<b>Programming Environments .....</b>	<b>1-5</b>
<b>1.6</b>	<b>Impact of Machine Architectures .....</b>	<b>1-6</b>
<b>1.7</b>	<b>Programming Paradigms .....</b>	<b>1-8</b>

**UNIT II****Chapter 2 : Structuring the Data, Computations & Program****2-1 to 2-30**

**Elementary Data Types** : Primitive data Types, Character String types, User Defined Ordinal Types, Array types, Associative Arrays, Record Types, Union Types, Pointer and reference Type.

**Expression and Assignment Statements** : Arithmetic expression, Overloaded Operators, Type conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed mode Assignment.

**Statement level Control Statements** : Selection Statements, Iterative Statements, Unconditional Branching.

**Subprograms** : Fundamentals of Sub Programs, Design Issues for Subprograms, Local referencing Environments, Parameter passing methods.

**Abstract Data Types and Encapsulation Construct** : Design issues for Abstraction, Parameterized Abstract Data types, Encapsulation Constructs, Naming Encapsulations.

<b>2.1</b>	<b>Introduction to Structuring of Data.....</b>	<b>2-1</b>
2.1.1	Elementary Data Types.....	2-1
2.1.2	Primitive Data Types.....	2-2
2.1.3	Character String Types.....	2-3
2.1.4	User Defined Ordinal Types .....	2-4
2.1.5	Array Types.....	2-5

2.1.6	Associative Arrays.....	2-6
2.1.7	Record Types.....	2-7
<b>2.2</b>	<b>Expressions and Statements .....</b>	<b>2-8</b>
2.2.1	Overloaded Operators.....	2-10
2.2.2	Type Conversions.....	2-11
2.2.3	Relational and Boolean Expression.....	2-11
2.2.4	Short-Circuit Evaluation.....	2-13
2.2.5	Assignment Statements .....	2-14
2.2.6	Selection Statements.....	2-18
2.2.7	Iterative Statements.....	2-19
2.2.8	Unconditional Branching.....	2-20
2.2.9	Subprograms .....	2-20
2.2.10	Design Issues for Subprograms.....	2-21
2.2.11	Local Referencing Environments.....	2-22
2.2.12	Parameter-Passing Methods.....	2-22
<b>2.3</b>	<b>Abstract Data Types and Encapsulation Construct.....</b>	<b>2-23</b>
2.3.1	Use of Static Variable and Methods .....	2-25
2.3.2	Design Issues for Abstraction.....	2-28
2.3.3	Parameterized Abstract Data Types.....	2-28
2.3.4	Encapsulation Constructs.....	2-28
2.3.5	Naming Encapsulations.....	2-29
➤	<b>Model Question Paper (In Sem.).....</b>	<b>M-1 to M-1</b>

### UNIT III

#### Chapter 3 : Java as Object Oriented Programming Language - Overview

3-1 to 3-57

**Fundamentals of JAVA, Arrays** : one dimensional array, multi-dimensional array, alternative array declaration statements, **String Handling** : String class methods, **Classes and Methods** : class fundamentals, declaring objects, assigning object reference variables, adding methods to a class, returning a value, constructors, this keyword, garbage collection, finalize() method, overloading methods, argument passing, object as parameter, returning objects, access control, static, final, nested and inner classes, command line arguments, variable - length arguments.

<b>3.1</b>	<b>Introduction.....</b>	<b>3-1</b>
3.1.1	What is Java?.....	3-1
3.1.2	Java History .....	3-2
3.1.3	Java Features .....	3-2

---

3.1.4	Java and Internet.....	3-4
<b>3.2</b>	<b>Java and Word Wide Web.....</b>	<b>3-4</b>
<b>3.3</b>	<b>Web Browsers .....</b>	<b>3-5</b>
<b>3.4</b>	<b>Java Virtual Machine .....</b>	<b>3-6</b>
3.4.1	JVM Architecture.....	3-7
<b>3.5</b>	<b>Data Types and Size.....</b>	<b>3-8</b>
3.5.1	Primitive Types .....	3-9
3.5.2	Non-Primitive Types.....	3-9
<b>3.6</b>	<b>Arrays.....</b>	<b>3-10</b>
3.6.1	One-Dimensional Array .....	3-10
3.6.2	Multi-dimensional Arrays.....	3-12
3.6.3	Alternative Array Declaration Statements .....	3-13
<b>3.7</b>	<b>String Handling : String Class Methods .....</b>	<b>3-18</b>
3.7.1	String Constructor.....	3-21
3.7.2	String Operations and Character Extraction .....	3-22
3.7.3	String Comparison .....	3-24
3.7.4	Searching in String.....	3-25
3.7.5	Modifying String.....	3-26
<b>3.8</b>	<b>Introduction to Classes and Methods.....</b>	<b>3-29</b>
<b>3.9</b>	<b>Class Fundamentals.....</b>	<b>3-29</b>
3.9.1	A Simple Class .....	3-30
3.9.2	Declaring Objects.....	3-31
3.9.3	Assigning Object Reference Variables .....	3-32
3.9.4	Introducing Methods.....	3-33
<b>3.10</b>	<b>Adding Methods to a Class .....</b>	<b>3-33</b>
<b>3.11</b>	<b>Returning a Value .....</b>	<b>3-34</b>
<b>3.12</b>	<b>Constructors .....</b>	<b>3-35</b>
3.12.1	Rules for Creating Java Constructor .....	3-36
3.12.2	Types of Java Constructors.....	3-36
<b>3.13</b>	<b>this Keyword.....</b>	<b>3-39</b>
<b>3.14</b>	<b>Garbage Collection.....</b>	<b>3-40</b>
<b>3.15</b>	<b>Finalize( ) Method.....</b>	<b>3-41</b>
<b>3.16</b>	<b>Overloading Methods .....</b>	<b>3-41</b>
<b>3.17</b>	<b>Argument Passing.....</b>	<b>3-44</b>
<b>3.18</b>	<b>Objects as Parameters.....</b>	<b>3-46</b>

---

<b>3.19 Returning Objects</b> .....	<b>3-47</b>
<b>3.20 Access Control</b> .....	<b>3-48</b>
3.20.1 Static Variables .....	3-49
3.20.2 Final Keyword .....	3-51
<b>3.21 Nested and Inner Classes</b> .....	<b>3-52</b>
<b>3.22 Command-Line Arguments</b> .....	<b>3-54</b>
<b>3.23 Variable-Length Arguments</b> .....	<b>3-55</b>

## UNIT IV

### Chapter 4 : Inheritance, Packages & Exception Handling Using Java

4-1 to 4-65

**Inheritances** : Member access and inheritance, super class references, Using super, multilevel hierarchy, constructor call sequence, method overriding, dynamic method dispatch, abstract classes, Object class.

**Packages and Interfaces** : Defining a package, finding packages and CLASSPATH, access protection, importing packages, interfaces (defining, implementation, nesting, applying), variables in interfaces, extending interfaces, instance of operator. fundamental, exception types, uncaught exceptions, try, catch, throw, throws, finally, multiple catch clauses, nested try statements, built-in exceptions, custom exceptions (creating your own exception sub classes).

**Managing I/O** : Streams, Byte Streams and Character Streams, Predefined Streams, Reading console Input, Writing Console Output, Print Writer class.

<b>4.1 Inheritances</b> .....	<b>4-1</b>
4.1.1 Member Access and Inheritance .....	4-1
4.1.2 Types of Inheritance .....	4-3
4.1.3 Advantages of Inheritance .....	4-14
<b>4.2 Method Overriding</b> .....	<b>4-16</b>
4.2.1 Polymorphism .....	4-17
<b>4.3 Dynamic Method Dispatch</b> .....	<b>4-18</b>
4.3.1 Advantages of Dynamic Method Dispatch .....	4-20
4.3.2 Difference between Static and Dynamic Dispatch .....	4-20
<b>4.4 Abstract classes</b> .....	<b>4-21</b>
4.4.1 Difference between Interface and Abstract Class .....	4-23
<b>4.5 Object Class</b> .....	<b>4-24</b>
<b>4.6 Packages and Interfaces</b> .....	<b>4-26</b>
4.6.1 Defining a Package .....	4-26
4.6.2 Finding Packages and Classpath .....	4-27
4.6.3 What is a Package ? .....	4-28

4.6.4	Access Protection .....	4-28
4.6.5	Importing Packages.....	4-29
4.6.6	Interfaces (Defining, Implementation, Nesting, Applying).....	4-29
4.6.7	Variables in Interface.....	4-32
4.6.8	Extending Interface .....	4-33
4.6.9	Instance of Operator.....	4-34
4.6.10	Differentiation between Class and Interface .....	4-34
<b>4.7</b>	<b>Introduction to Exception Handling.....</b>	<b>4-36</b>
4.7.1	Advantages of Exception Handling.....	4-37
<b>4.8</b>	<b>Exception Handling Fundamentals.....</b>	<b>4-38</b>
4.8.1	Exception Types.....	4-39
4.8.2	Uncaught Exceptions .....	4-39
4.8.3	Try and Catch Blocks .....	4-40
4.8.4	Throw Block.....	4-41
4.8.5	Throws .....	4-42
4.8.6	Finally.....	4-43
4.8.7	Multiple Catch Clauses .....	4-44
4.8.8	Nested Try Statements.....	4-46
4.8.9	Built-in Exceptions.....	4-47
4.8.10	Custom Exception.....	4-48
<b>4.9</b>	<b>Managing I/O .....</b>	<b>4-49</b>
4.9.1	Streams .....	4-50
4.9.2	ByteStream.....	4-51
4.9.3	CharacterStream.....	4-57
4.9.4	Predefined Streams .....	4-60
<b>4.10</b>	<b>Reading Console Input .....</b>	<b>4-60</b>
<b>4.11</b>	<b>Writing Console Output .....</b>	<b>4-61</b>
4.11.1	Print Writer Class.....	4-62

## UNIT V

### Chapter 5 : Multithreading in Java

5-1 to 5-24

**Concurrency and Synchronization, Java Thread Model :** Thread priorities, Synchronization, Messaging, Main Thread, Creating thread: Implementing Thread using thread class and Runnable interface. Creating multiple threads using is Alive() and join(). **Web Based Application in Java :** Use of JavaScript for creating web based applications in Java, Introduction to Java script frameworks- ReactJS, VueJS, AngularJS (open source).

<b>5.1</b>	<b>Introduction</b> .....	<b>5-1</b>
<b>5.2</b>	<b>Java Thread Model</b> .....	<b>5-2</b>
5.2.1	Types of Java Thread Model.....	5-2
5.2.1(A)	Thread Priorities.....	5-3
5.2.1(B)	Synchronization.....	5-3
5.2.1(C)	Messaging.....	5-3
<b>5.3</b>	<b>The Main Thread</b> .....	<b>5-3</b>
<b>5.4</b>	<b>Creating a Thread</b> .....	<b>5-6</b>
<b>5.5</b>	<b>Executing Runnable</b> .....	<b>5-7</b>
<b>5.6</b>	<b>Creating Multiple Threads</b> .....	<b>5-9</b>
<b>5.7</b>	<b>Web based Application in Java</b> .....	<b>5-13</b>
5.7.1	Building a Webapp.....	5-15
<b>5.8</b>	<b>Introduction to Java Script Frameworks- ReactJS, VueJS, AngularJS (Open Source)</b> .....	<b>5-20</b>
5.8.1	ReactJS.....	5-20
5.8.2	VueJS.....	5-21
5.8.3	AngularJS.....	5-21

## UNIT VI

### Chapter 6 : Logical & Functional Programming

**6-1 to 6-32**

**Functional Programming Paradigm** : Understanding symbol manipulation, Basic LISP functions, definitions, predicates, conditionals and scoping, Recursion and iteration, Properties List array and access functions, Using lambda definitions, printing, reading and atom manipulation.

**Logic Programming Paradigm** : An Overview of Prolog, Syntax and Meaning of Prolog Programs, Lists, Operators, Arithmetic, Using Structures.

<b>6.1</b>	<b>Introduction</b> .....	<b>6-1</b>
6.1.1	Functional Programming Paradigm.....	6-2
6.1.2	Functional Programming - Characteristics.....	6-2
6.1.3	Feature of Functional Paradigm.....	6-2
6.1.4	Advantages of Functional Programming.....	6-4
6.1.5	Disadvantages of Functional Programming.....	6-4
<b>6.2</b>	<b>Understanding Symbol Manipulation</b> .....	<b>6-4</b>
<b>6.3</b>	<b>Basic LISP Functions</b> .....	<b>6-5</b>
6.3.1	Features of Common LISP.....	6-5
6.3.2	Implementations.....	6-5

---

<b>6.4</b>	<b>LISP Uses Prefix Notation</b> .....	<b>6-6</b>
<b>6.5</b>	<b>Global Variables</b> .....	<b>6-9</b>
<b>6.6</b>	<b>Local Variables</b> .....	<b>6-9</b>
<b>6.7</b>	<b>LISP Constants</b> .....	<b>6-10</b>
<b>6.8</b>	<b>Defining Functions in LISP</b> .....	<b>6-11</b>
<b>6.9</b>	<b>Predicates in LISP</b> .....	<b>6-12</b>
<b>6.10</b>	<b>Conditionals in LISP</b> .....	<b>6-14</b>
<b>6.11</b>	<b>Scoping using LISP</b> .....	<b>6-15</b>
6.11.1	Lexical Scoping Examples .....	6-16
6.11.2	Dynamic Scoping .....	6-16
6.11.3	Problems with Dynamic Scope .....	6-17
<b>6.12</b>	<b>Recursion in LISP</b> .....	<b>6-17</b>
<b>6.13</b>	<b>Lists in LISP</b> .....	<b>6-19</b>
6.13.1	List Manipulating Functions.....	6-19
6.13.2	Concatenation of car and cdr Functions.....	6-21
<b>6.14</b>	<b>Access Function</b> .....	<b>6-21</b>
<b>6.15</b>	<b>Lambda Function</b> .....	<b>6-22</b>
<b>6.16</b>	<b>Atoms in LISP</b> .....	<b>6-22</b>
<b>6.17</b>	<b>Logic Programming : ProLog</b> .....	<b>6-23</b>
6.17.1	Data Types .....	6-23
6.17.2	Comparison Operators.....	6-24
6.17.3	Arithmetic Operators in Prolog.....	6-25
6.17.4	Recursion .....	6-26
6.17.5	Matching in Prolog.....	6-28
➤	<b>Model Question Paper (End Sem.)</b> .....	<b>M-1 to M-2</b>

---