Object Oriented Analysis and Design using Java

UE19CS353

Prof . Sudeepa Roy Dey and Prof. Sindhu R Pai
Department of Computer Science and Engineering

PES UNIVERSITY ONLINE

UE19CS353: Object Oriented Analysis and Design using Java

Overview

Prof . Sudeepa Roy Dey and Prof. Sindhu R Pai

Department of Computer Science and Engineering

Agenda

- Syllabus
- Text Books
- Evaluation Policy
- Introduction to course
- What we know?
- What we will study?
- Software Evolution – Procedural, OO and OB Programming
- Introduction to JAVA
- Features of JAVA
- Main inter-related components of JAVA
- References

Syllabus

Unit 1: Introduction to Object Oriented Programming

Unit 2: Advanced OO, Object Oriented Analysis and Static Models and Diagrams

Unit 3: Dynamic Models, Diagrams and Architecture design and principles

Unit 4: OO Design Principles and Sample Implementation of Patterns in Java

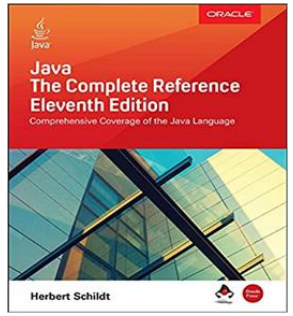
Unit 5: OO Design Patterns & Anti-Patterns with Sample implementation in Java

Object Oriented Analysis and Design with Java

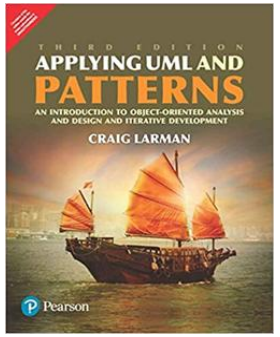
Text books and Reference Books

Text Books:

1:"Java the Complete Reference", Herbert Schildt ,McGraw-Hill ,11th Edition, 2018.



2: "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", by Craig Larman, 3rd Edition, Pearson 2015.



References:

1: "Object-Oriented Modelling and Design with UML", Michael R Blaha and James R Rumbaugh, 2nd Edition, Pearson 2007.

2: "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, 1st Edition, Pearson 2015.

Object Oriented Analysis and Design with Java

Evaluation Policy

ISA	ISA 1 (First 2 Units) – CBT	TBD
	ISA 1 (Next 3 Units) – CBT	TBD
	Assignment (1 per Unit)	TBD
	Project	TBD
ESA	Pen and Paper Mode (100 marks)	50 marks

Course Outcome

Unit 1: Introduction to Object Oriented Programming

CO1: Demonstrate ability to understand the various concepts of object-oriented programming language using Java.

Unit 2: Object Oriented Analysis and Static Models and Diagrams

CO 2 &3: Understanding and designing various static UML diagrams .

Unit 3: Dynamic Models, Diagrams and Architecture design and principles.

CO4: Implementing Dynamic UML models, Grasp principles

Unit 4: OO Design Principles and Sample Implementation of Patterns in Java

CO4: Implementing Various patterns, SOLID principles in Java.

Unit 5: OO Design Patterns & Anti-Patterns with Sample implementation in Java

CO5: Demonstrate and appreciate Patterns and Anti-patterns

Introduction

- Two main parts:

  Object Oriented Analysis (OOA)

  Object Oriented Design (OOD)

- **Object Oriented Analysis (OOA)**

  The first technical activity performed as part of object oriented software engineering.

  Introduces new concepts to investigate a problem.

  Based on a set of **basic principles**:

  The information domain is modeled. Behavior is represented.

  Function is described. Data, functional, and behavioral models are divided.

  Early models represent the essence of the problem, while later ones provide implementation

Introduction

- Two main parts:

  Object Oriented Analysis (OOA)

  Object Oriented Design (OOD)

- **Object Oriented Design (OOD)**

An analysis model created using OOA is transformed by OOD into a design model that works as a plan for software creation.

Results in a design having several different levels of modularity

The major system components are partitioned into subsystems

Data manipulation operations are encapsulated into objects

Must specify some data organization of attributes and a procedural description of each operation

Object Oriented Design Pyramid

The Subsystem Layer :

Represents the subsystem that enables software to achieve user requirements and implement technical frameworks that meet user needs.

The Class and Object Layer :

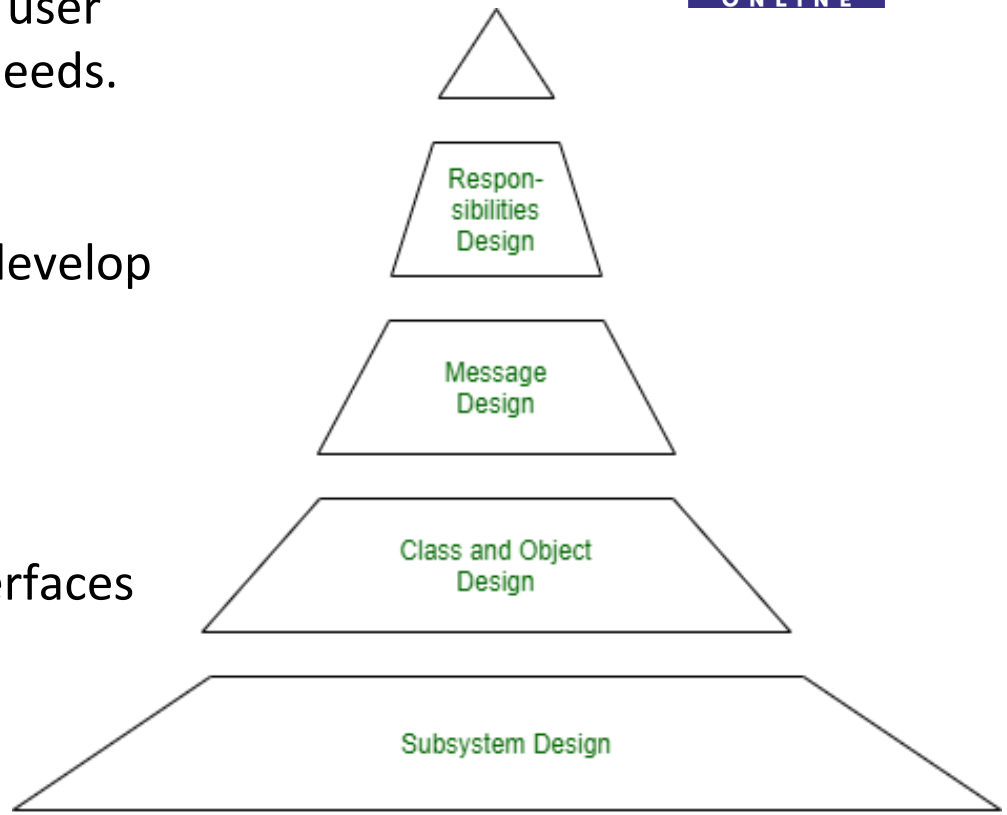
Represents the class hierarchies that enable the system to develop using generalization and specialization. Also represents each object.

The Message Layer :

Represents the design details that enable each object to communicate with its partners. Establishes internal and external interfaces for the system.

The Responsibilities Layer :

Represents the data structure and algorithmic design for all the attributes and operations for each object.



The Object Oriented Design Pyramid

What we know?

- ## Software Development Life Cycle(SDLC)

  Produces software with highest quality and lowest cost in the shortest time possible.

  Phases involved - **Requirement Analysis, Planning, Design, Development, Testing and Deployment**

  The design step falls between understanding the requirements and building the product

  Popular models - **waterfall, spiral and agile**

- ## Structured Systems Analysis & Design(SSAD)

  A set of standards for systems analysis and application design.

  It uses a formal methodical approach to the analysis and design of information systems.

  Developed by Learmonth Burchett Management Systems (LBMS) and the Central Computer Telecommunications Agency in 1980-1981 as a standard for developing British database projects.

  An open methodology based on the waterfall model.

  Used by commercial businesses, consultants, educational establishments and CASE tool developers.

What will we study?

- **Object Oriented Analysis and Design (OOAD)**

  One approach to make the design process easier is the object-oriented(OO) approach

  This allows for the description of **concepts** in the problem and solution spaces as **objects**

- **What is Object Oriented Approach?**

  In software development lifecycle we can apply and implement OO concepts by following three steps. **OO Analysis --> OO Design --> OO implementation by using OO languages(Java)**

Context of Object-oriented Approach

Software Crisis

Developments in software technology continue to be dynamic.

New tools and techniques are announced in quick succession.

This has forced the software engineers and industry to continuously look for new approaches to software design and development.

These rapid advances appear to have created a situation of crisis within the industry

The following issues need to be addressed to face the crisis:

How to represent real-life entities of problems in system design?

How to design system with open interfaces?

How to improve the quality of software, software productivity and decrease software cost?

How to ensure reusability and extensibility of modules?

How to develop modules that are tolerant of any changes in future?

Software Evolution

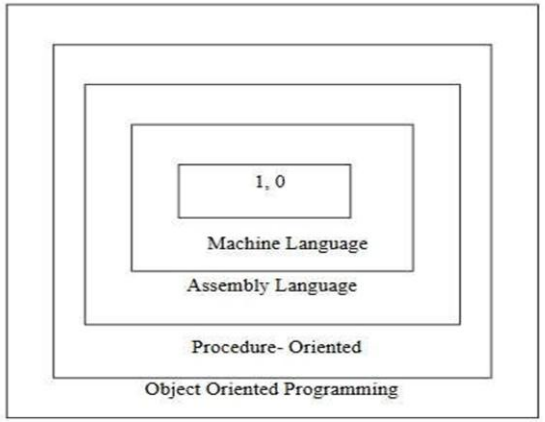
- Process of developing software initially, then timely updating it for various reasons

i.e., add new features or to remove obsolete functionalities.

- **Phases:**

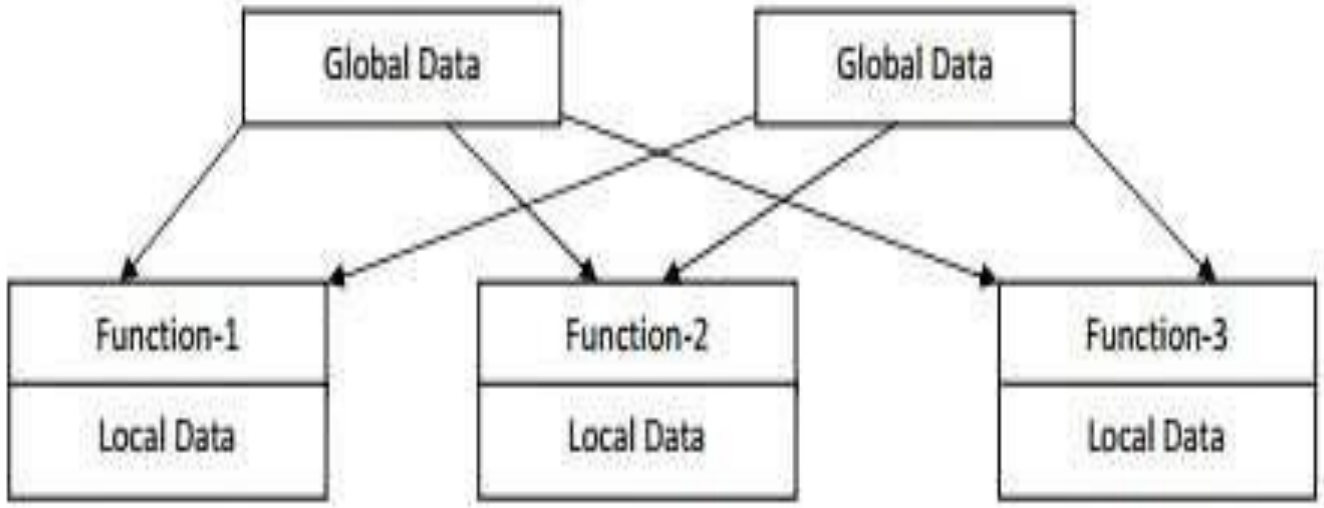
Initial Development, Evolution, Servicing, Phase-out, close-down

- **Programming knowledge**



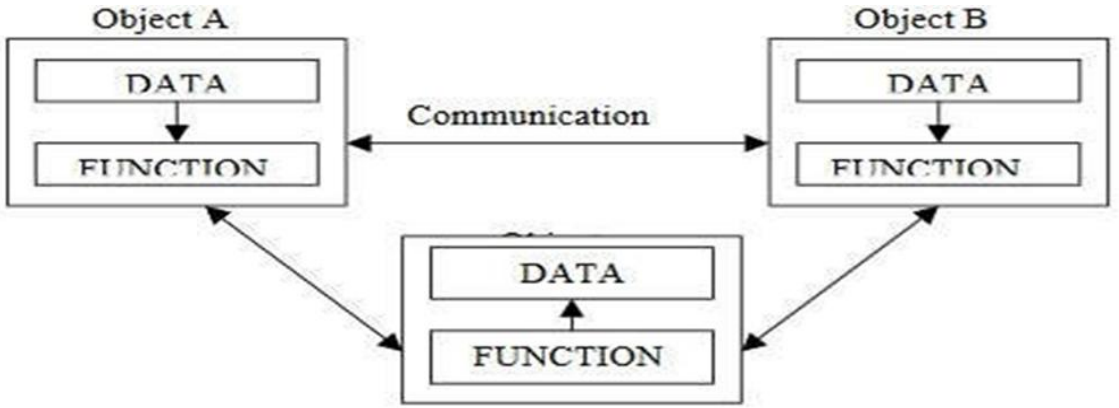
Procedural programming

- Conventional programming, using languages such as COBOL, FORTRAN and C
- The problem is viewed as a sequence of things to be done such as reading, calculating and printing
- A number of functions are written to accomplish these tasks .The primary focus is on **functions**



Object-oriented Programming

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data is hidden and cannot be accessed by external functions.
- Objects may communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- Follows bottom-up approach in program design



Object - Oriented vs Object - Based Programming

Object-oriented Programming Language	Object-based Programming Language
All the characteristics and features of object-oriented programming are supported - **abstraction, encapsulation, inheritance, polymorphism** Ex: Java, python	All characteristics and features of object-oriented programming are supported **except for inheritance and polymorphism** Ex: Javascript, VB

Introduction to JAVA

- Originally developed at **Sun Microsystems** by **James Gosling** and **released in 1995** as core component of Sun Microsystems' Java platform (Java 1.0)
- The **new J2 versions** were renamed as **Java SE, Java EE and Java ME** respectively.

  Java Platform, Standard Edition (Java SE)

  Java Platform, Enterprise Edition (Java EE)

  Java Platform, Micro Edition (Java ME)
- Java is guaranteed to be **Write Once, Run Anywhere.**
- Java was mainly developed to create software for consumer electronic devices that could be controlled by a remote.

Features of JAVA

1. **Simple:** No header files, pointer arithmetic, structures, unions, operator overloading, virtual base classes

2. **Object-Oriented:** Focuses on the data (= objects) and on the interfaces to the object.

3. **Distributed:**

   Has an extensive library of routines for coping with TCP/IP protocols like HTTP and FTP.

   Applications can open and access objects across the Net via URLs with the same ease as when accessing a local file system

4. **Robust:** Inbuilt exception handling features and memory management features

5. **Secure:**

   Intended to be used in networked/distributed environments.

   Toward that end, a lot of emphasis has been placed on security.

   Java enables the construction of virus-free, tamper-free systems

Features of JAVA continued..

6. Portable:

Unlike C and C++, there are no “implementation-dependent” aspects of the specification.

The sizes of the primitive data types are specified as is the behavior of arithmetic on them.

7. Interpreted: Programmer writes code that will be executed by an interpreter

8. High-Performance:

The byte codes can be translated on the fly (at runtime) into machine code for the particular CPU the application is running on.

The just-in-time compiler knows which classes have been loaded.

It can use inlining based upon the currently loaded collection of classes, a particular function

is never overridden, and it can undo that optimization later if necessary.

Main inter-related components of JAVA

- JAVA Development Kit (JDK)
- JAVA Runtime Environment (JRE)
- JAVA Virtual Machine (JVM)

JAVA Development Kit (JDK)

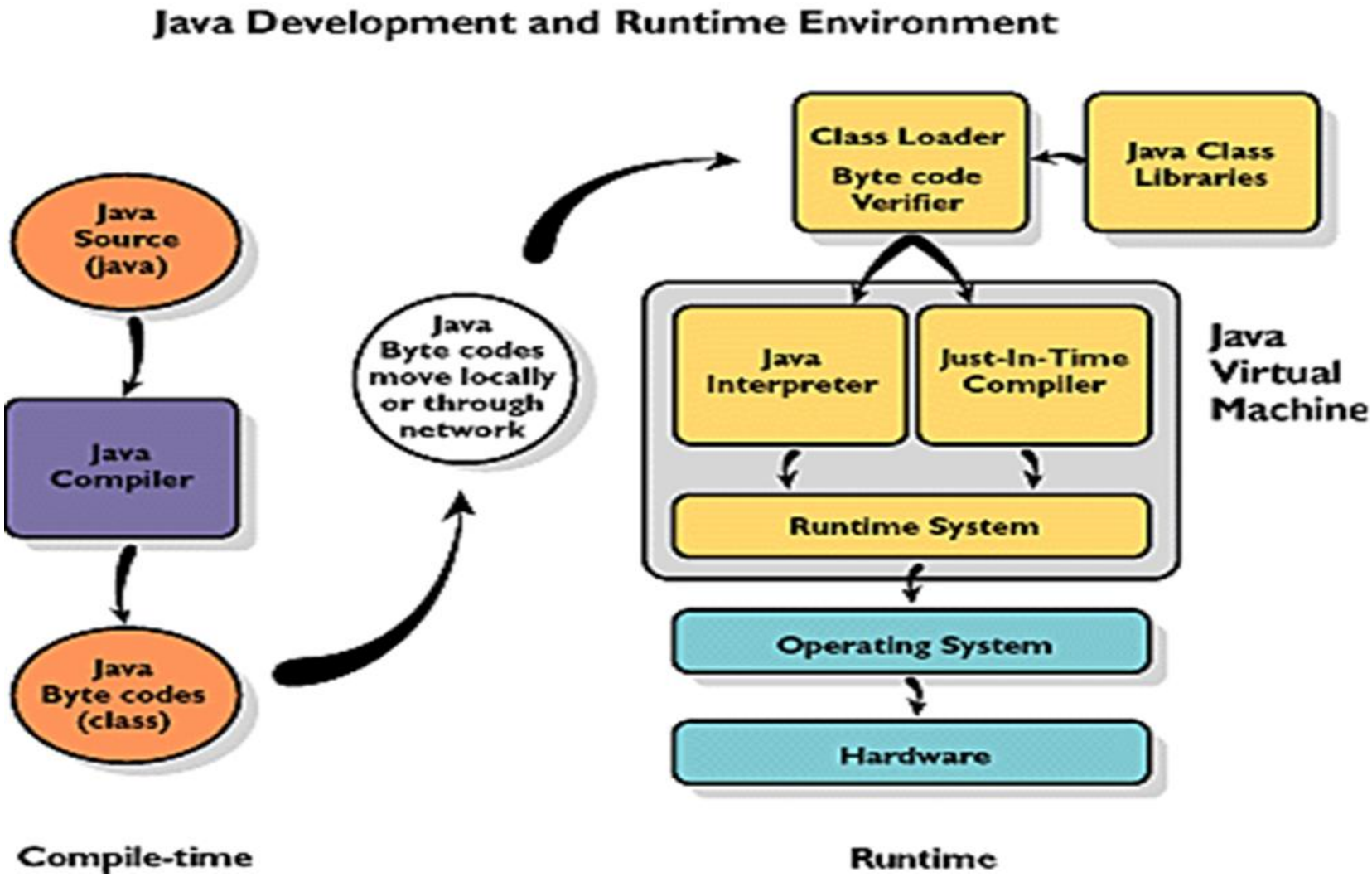
- A set of tools/software package for developing/creating Java applications

- Implementation of the Java platform specification, including compiler and class libraries

- Developers choose JDKs **by Java version and by package or edition**

- Every JDK always includes a compatible JRE, because running a Java program is part of the process of developing a Java program

Java Runtime Environment(JRE)



JAVA Virtual Machine

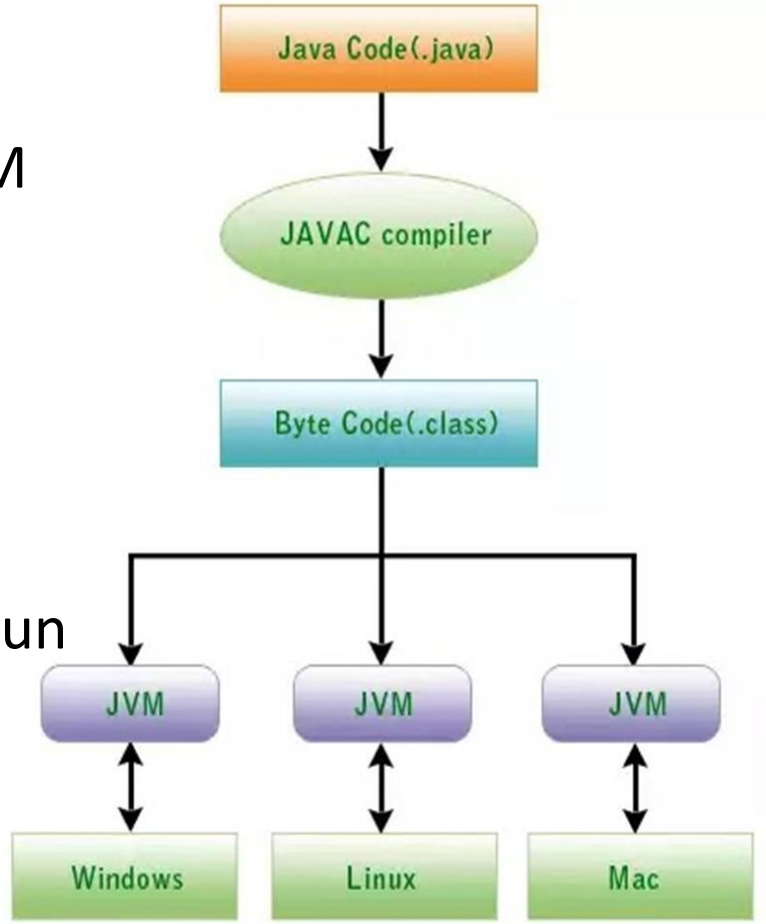
- Two primary functions:

  To allow Java programs to run on any device or operating system

  To manage and optimize program memory

- Once the class loader has done its work of loading classes, the JVM begins executing the code in each class. The execution engine is the JVM component that handles this function. The execution engine is essential to the running JVM.

- Whenever you write java command on the command prompt to run the java class, an instance of JVM is created



References

1.“Java the Complete Reference”, Herbert Schildt ,McGraw-Hill ,11th Edition, 2018.

2. “Object-Oriented Modelling and Design with UML”, Michael R Blaha and James R Rumbaugh, 2nd Edition, Pearson 2007.

3. “Design Patterns: Elements of Reusable Object-Oriented Software” by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, 1st Edition, Pearson 2015

4. Web References : https://www.javatpoint.com/jvm-java-virtual-machine

5. Object Oriented Analysis and Design – GeeksforGeeks

6. What Is SDLC? Understand the Software Development Life Cycle – Stackify

7. What is the JRE (Java Runtime Environment)? - India | IBM

THANK YOU

Sudeepa Roy Dey and Sindhu R Pai
Department of Computer Science and Engineering

sudeepar@pes.edu

sindhurpai@pes.edu
+918277606459