



Object Oriented Analysis and Design with Java

UE19CS353

Prof . Sudeepa Roy Dey

Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

UE19CS353: Object Oriented Analysis and Design with Java

Overview

Prof Sudeepa Roy Dey

Department of Computer Science and Engineering

Object Oriented Analysis and Design with Java

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

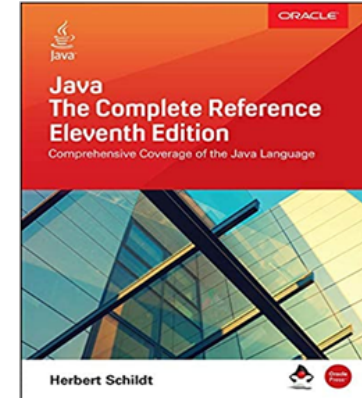
Unit 4: OO Design Principles and Sample Implementation of Patterns in Java

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

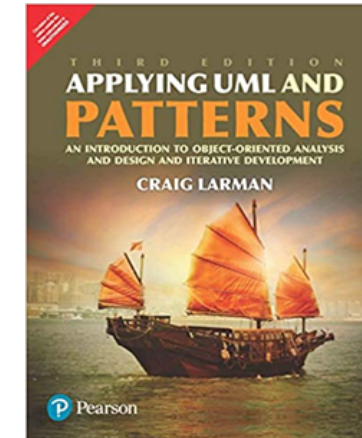
Object Oriented Analysis and Design with Java

Text and Reference Books

T1: “Java the Complete Reference”, Herbert Schildt ,McGraw-Hill ,11th Edition, 2018.



T2: “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	ISAs – CBT	TBD
	Assignment (1 per Unit)	TBD
	Project	TBD
ESA	Pen and Paper Mode (100 marks)	50 marks

Object Oriented Analysis and Design with Java

What we Know?



Traditional SDLC

- In Software Engineering course, the design step falls between understanding your requirements and building the product.

Structured Systems Analysis & Design(SSAD)

Structured Systems Analysis & Design (SSAD) is a framework of activities and tasks that need to be accomplished to develop an information system. This approach is also known as top-down design, modular programming, and stepwise refinement.

Object Oriented Analysis and Design with Java

What will we study?



What is OOAD?

One approach to help 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)

Object Oriented Analysis and Design with Java



Course Outcome

Unit : 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

Object Oriented Analysis and Design with Java

Context of Object-oriented Approach



Software Crisis

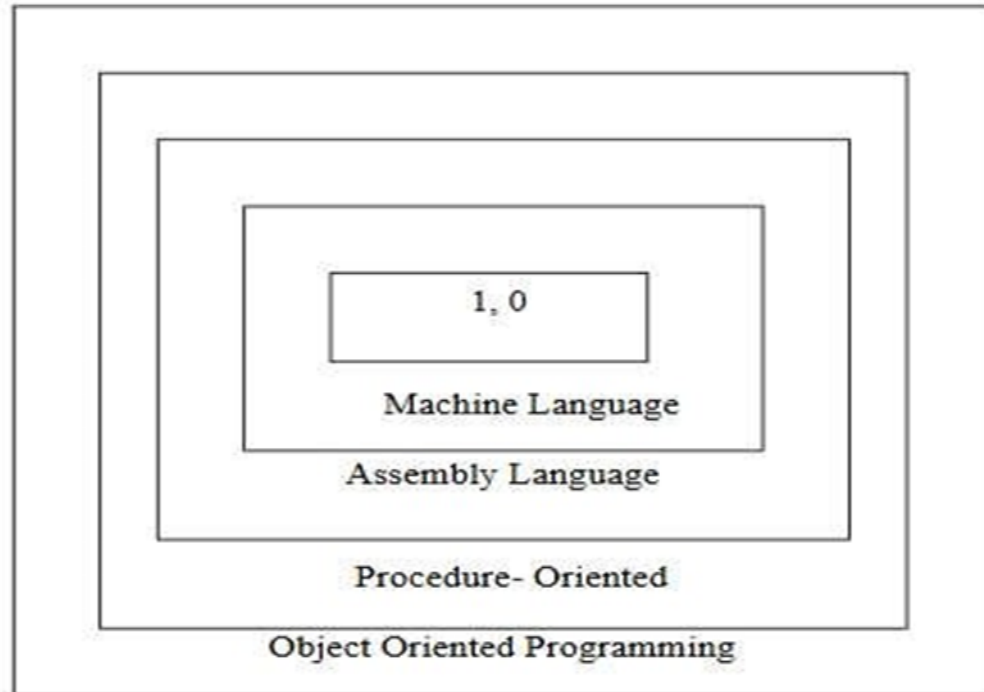
- ❑ Developments in software technology continue to be dynamic.
- ❑ New tools and techniques are announced in quick succession.
- ❑ This has forced the software engineers and industry to continuously look for new approaches to software design and development.
- ❑ These rapid advances appear to have created 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?
- How to **ensure reusability and extensibility** of modules?
- How to develop modules that are tolerant of any changes in future?
- How to improve software productivity and decrease software cost?

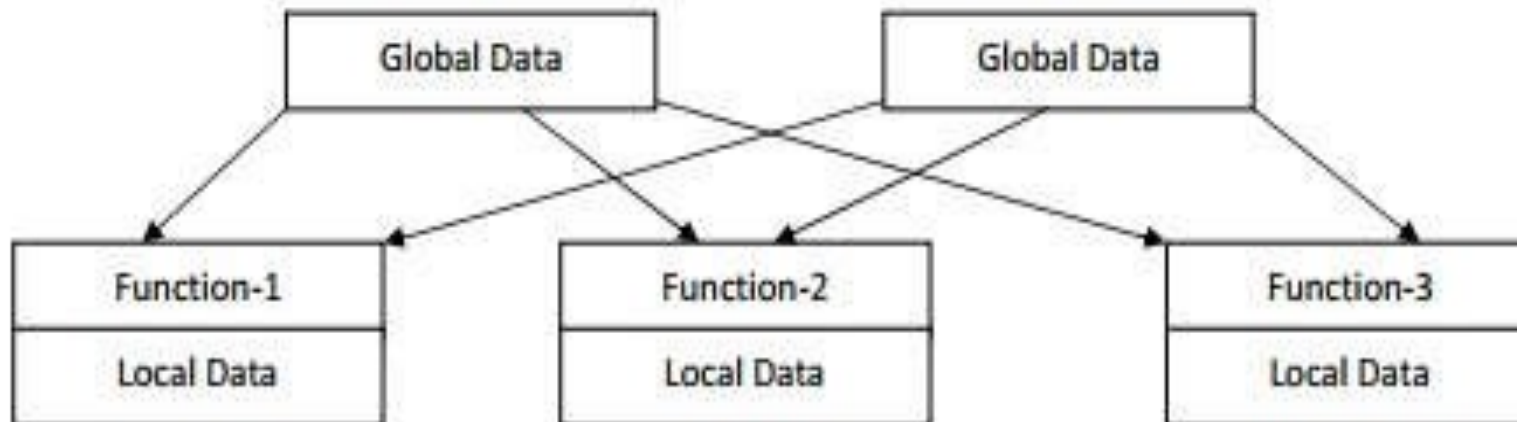
Object Oriented Analysis and Design with Java

Software Evolution concepts: Programming Knowledge



Procedure-oriented programming

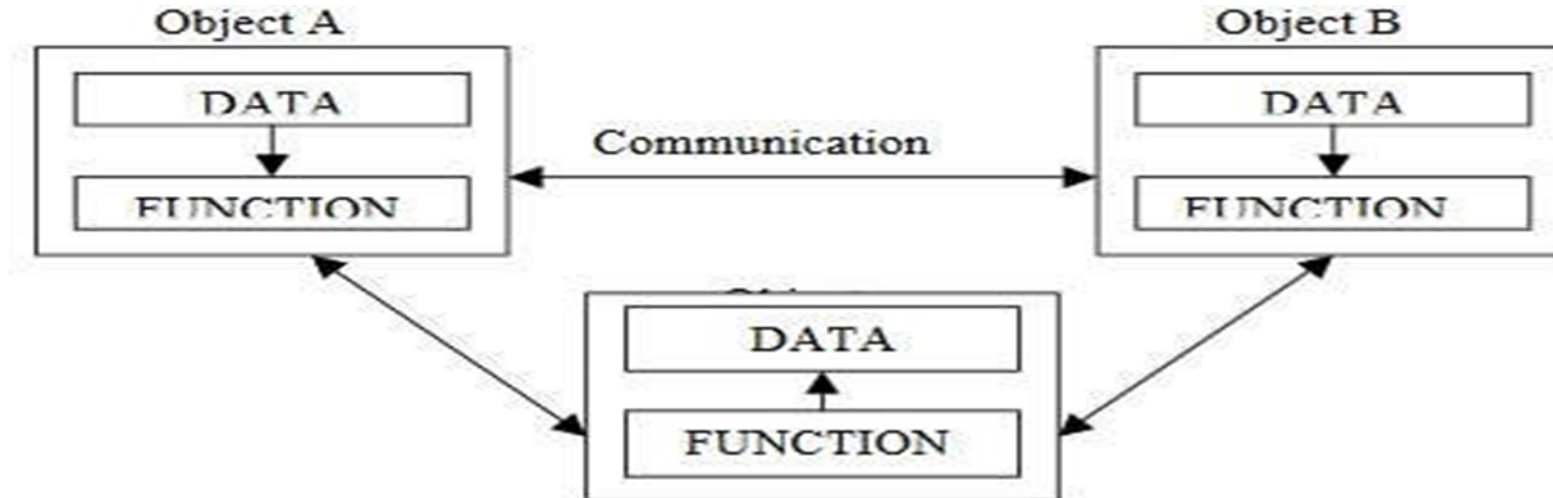
- ❑ Conventional programming, using high level languages such as **COBOL, FORTRAN** and **C**, is commonly known as **procedure-oriented programming (POP)**.
- ❑ In the procedure-oriented approach, the problem is viewed as a **sequence of things to be done** such as reading, calculating and printing. A number of functions are written to accomplish these tasks.
- ❑ The primary focus is on **functions** :



Object Oriented Analysis and Design with Java

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 Analysis and Design with Java

Object-based Programming

Object-oriented Programming Language	Object-based Programming Language
All the characteristics and features of object-oriented programming are supported.	All characteristics and features of object-oriented programming, such as inheritance and polymorphism are not supported.
These types of programming languages don't have a built-in object. Example: C++.	These types of programming languages have built-in objects. Example: JavaScript has a window object.
Java is an example of object-oriented programming language which supports creating and inheriting (which is reusing of code) one class from another.	VB is another example of object-based language as you can create and use classes and objects, but inheriting classes is not supported.

Object Oriented Analysis and Design with Java

Introduction to Java



- ❑ Java programming language was originally developed by 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.
 1. Java Platform, Standard Edition (Java SE)
 2. Java Platform, Enterprise Edition (Java EE)
 3. 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.

Object Oriented Analysis and Design with Java

Introduction to Java: Features



1. Simple

There is no need for header files, pointer arithmetic, structures, unions, operator overloading, virtual base classes

2. Object-Oriented

Object-oriented design is a programming technique that focuses on the data (= objects) and on the interfaces to that object.

3. Distributed

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

Java is 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.

Introduction to Java: Features



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**, rather than compiled into object code loaded by the OS and executed by CPU directly. An interpreter executes the code line by line.

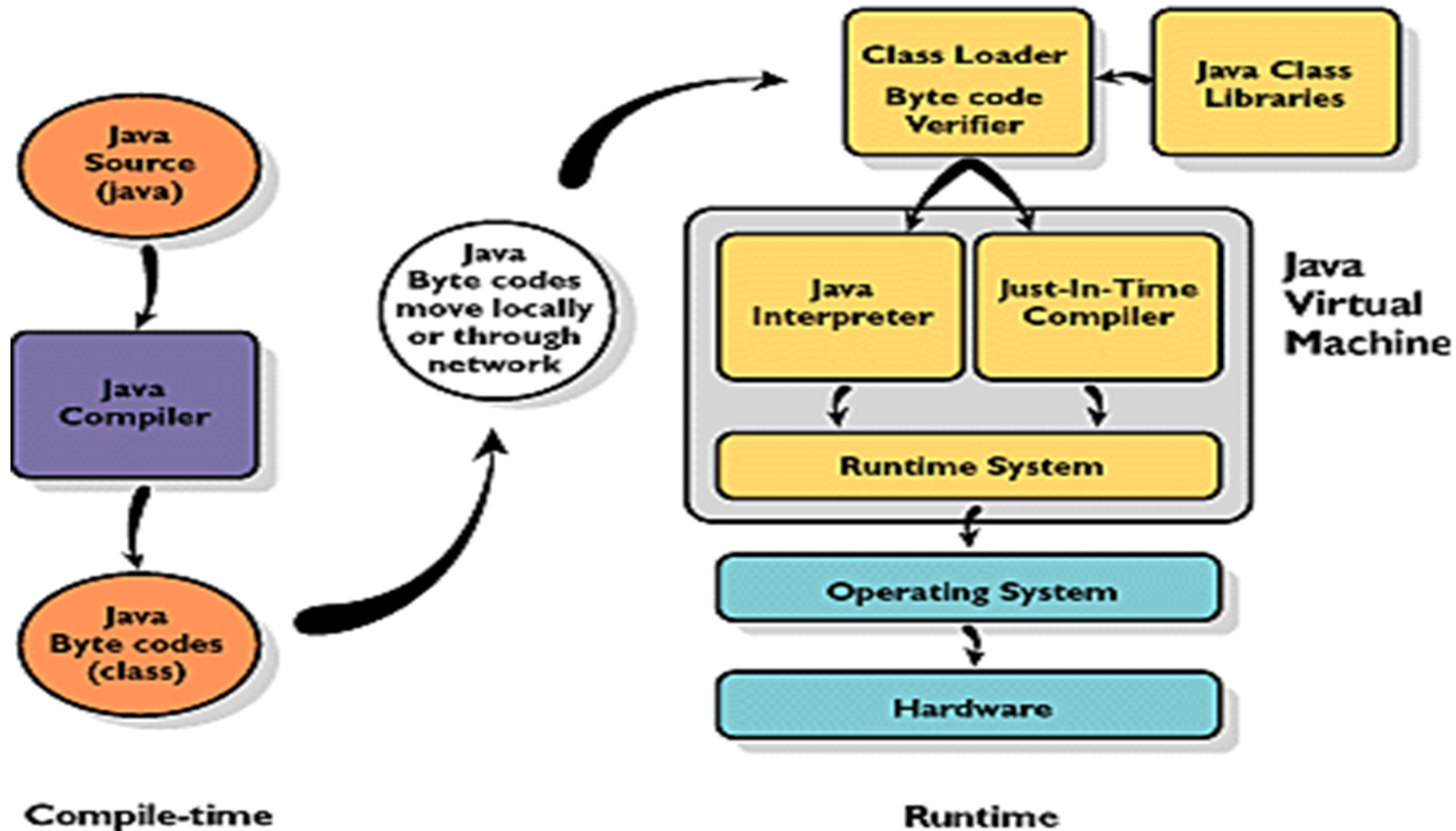
8. High-Performance

The bytecodes 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 when, based upon the currently loaded collection of classes, a particular function is never overridden, and it can undo that optimization later if necessary.

Object Oriented Analysis and Design with Java

Java Runtime Environment(JRE)

Java Development and Runtime Environment



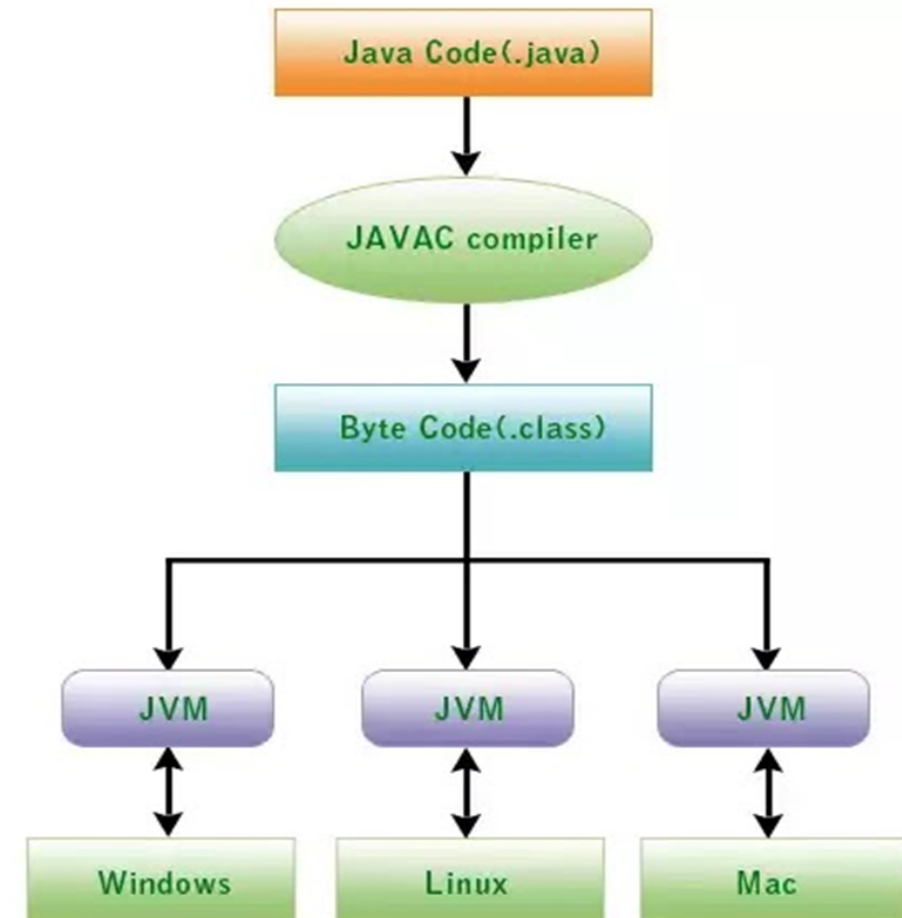
Object Oriented Analysis and Design with Java

JAVA Virtual Machine

JVM is a **platform-independent execution environment** that converts Java bytecode(.class file) into machine language and executes it.

It is:

- **A specification** where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.
- **An implementation** Its implementation is known as JRE (Java Runtime Environment).
- **Runtime Instance** Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.



Object Oriented Analysis and Design with Java

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>



THANK YOU

Sudeepa Roy Dey

Department of Computer Science and Engineering

sudeepar@pes.edu