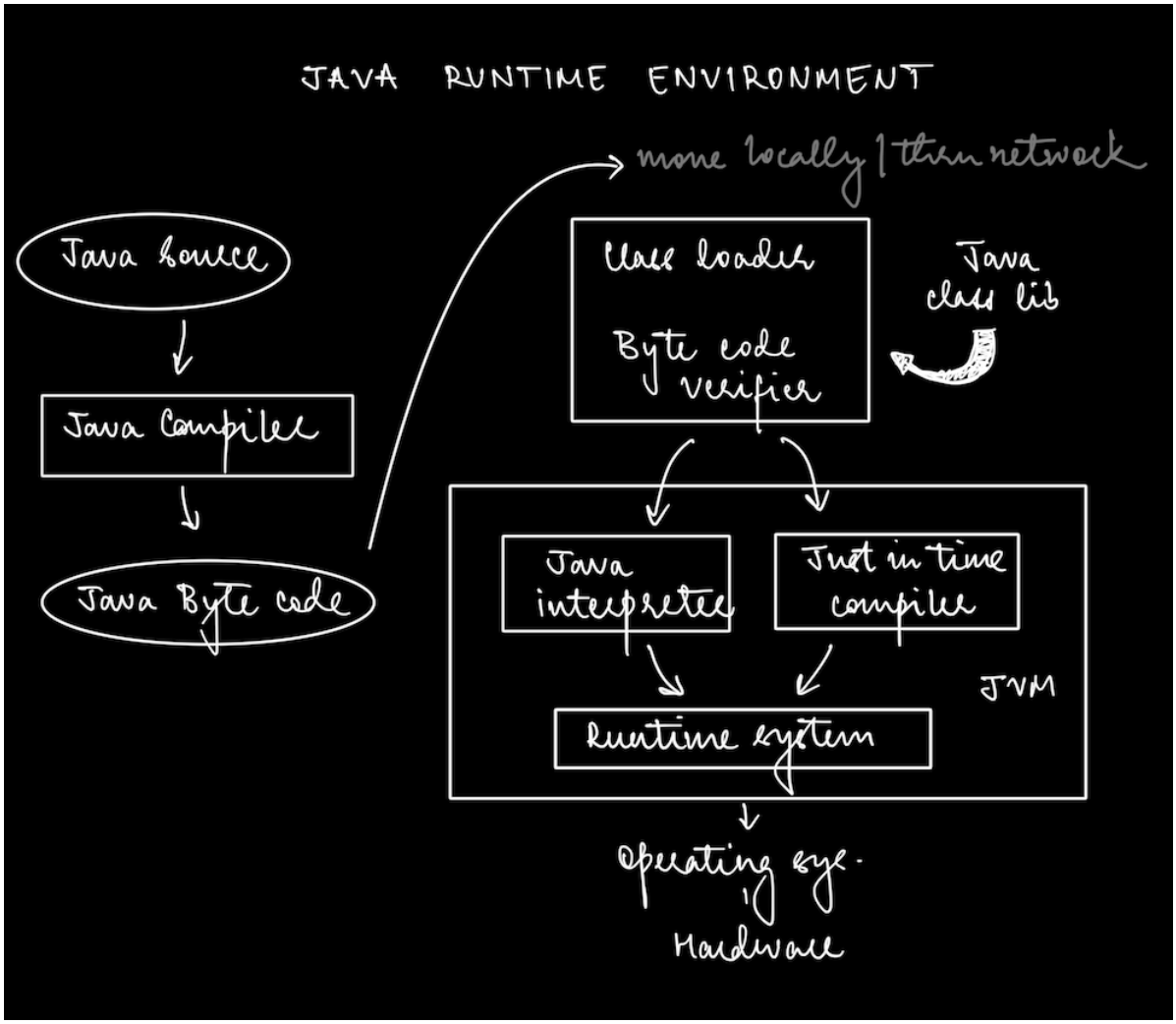
Objected Oriented Analysis and Design with Java - II

UNIT II - Object Oriented Programming

▼ Java Runtime Environment



Java Virtual Machine

JVM is a platform independent execution environment that converts Java bytecode (.class) file into machine language and then executes it. Its implementation is known as Java runtime environment.

▼ Java Concepts

- **Encapsulation** - The wrapping up of data and functions into a single unit. These functions provide the interface between the object's data and the program.
- **Data Abstraction** - Representing essential features without including the background details or explanation.
- **Composition and Inheritance** - A way of representation "has a" relationship and an "is a" relationship.

▼ Access Modifiers

- Private - within the class
- Protected - within package and outside the package only through child class
- Public - can be accessed from anywhere
- Default - only within the package

→ Accessing private objects outside of their class results in compile time error

▼ Constructors

A constructor initializes an object when called. Syntactically similar to a method and has no explicit return type.

1. **Default constructor** - Provides default values to objects, like 0, false, null, depending on the datatype of the instance variables.

2. **Parameterized constructor** - Initializes fields of the object with the given value. There is no return value but it returns the current class instance.
3. **Copy constructor** - Used to create an exact copy of the existing object.

▼ Garbage collector

Remove unused objects from memory and free space.

Automatic garbage collection - *Garbage collector* - program running in the background that looks into all objects in the memory and finds out objects that aren't referenced by any part of the program. These objects are deleted and space is reclaimed for allocation to other objects.

Finalization - Certain actions need to be performed before an object is destroyed, like closing all db connections and files, releasing network resources, and recovering heap space and the release of release locks. Java provides a mechanism called finalization through the finalize() method. Called prior to garbage collection.

Java calls this method whenever it's about to recycle an object of a class.

▼ Parameter passing

Java is strictly pass by value

There are two types of parameters -

1. Formal parameter
    - Any modifications to the formal parameter variable inside the function affect only the separate storage location and will not be reflected in the actual parameter in the calling environment.
    - Changes made to formal parameter get transmitted back to the caller through parameter passing.
2. Actual parameter

▼ Method Overloading

Allows a class to have more than one method with the same name if their argument lists are different.

- number of parameters vary

- data type of parameters vary
- order of parameters of methods vary

▼ Method types

- Instance methods - Object of its class should be created before it's called.
- Static methods - Referenced by class name or reference to object of that class. The Static keyword can also be used for for a variable, a method and a block.

  - **Class variables - Static fields**
    → Variables that are common to all objects.
    → They are associated with the class rather than the object.
    → Every instance of a class shares a class variable, which is in one fixed location in memory
  - **Static block**
    → Used to initialize the static data member. Executed before main method at the time of class loading.

▼ Method Overriding

If a subclass provides the specific implementation of a method which is been declared by a parent class, it is called as method overriding.

→ Also called as run time polymorphism.

▼ Inheritance

Types of inheritance -

1. Single level inheritance
2. Multilevel inheritance
3. Hierarchical inheritance

▼ Interfaces

An interface is the blueprint of a class. It specifies what a class can do. Methods declared in an interface are by default abstract.

Data members in an interface will be for the whole class and will be immutable.

▼ Abstract class

Provides implementation reuse - default implementation.

To create a superclass that only defines a generalized form that will be shared by all its subclasses, leaving it to each subclass to fill the details.

Referred to as subclasser responsibility as they have no implementation specified in the super class.

▼ Difference between an Abstract class and an Interface

Speed	Fast	Relatively slow
Multiple Inheritance	Extends only one class	Can implement multiple interfaces
Defined fields	Define fields and constants	Cannot define fields
Extension limit	A single abstract class can extend one and only one interface	A single interface can extend multiple interface.

▼ Object class

Object class defined by java is a super class of all other classes, in the absence of any other explicit superclass. Defined in the java.lang package

▼ Array, List and Stack

Contiguous allocation of memory.

→ Foreach loop syntax:

```
for(data_type variable:array){}
```