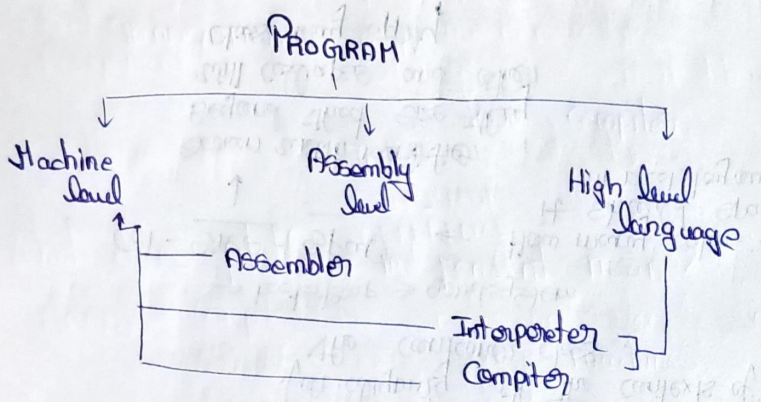
WEEK-1

JAVA INTRODUCTION
#LECTURE-1



→ 3rd Generation Programming language (3GL).

→ Higher level lang can be distinguished by

* Function-oriented programming.
* object-oriented programming.

PROGRAM WRITTEN BASED ON:

* Multiple function (set of func)
* Different/share data.

Eg: C.

* There is a global database.

* States of object
* Set of objects.
* There is no global database. Data stored within different object.

FOP VS OOP

	FOP	OOP
PROGRAM DIVIDED INTO PARTS CALLED	functions	objects.
	top down approach	bottom up approach.
	No access specifiers	3 ⇒ public, private, protected.
	Polymorphism not possible	Polymorphism is possible.

*JAVA PROGRAMING PARADIGM

→ Encapsulation → Information hiding "ACCESS MODIFIERS"
→ Inheritance
→ Polymorphism.

Encapsulation: defining a class

Contains Variables, data. methods.

They put together define the class.

After a class is created, then we can create object.

Inheritance: object acquires all the properties and behaviours of parent object.

Polymorphism: ability to process object depending on their class.

Same name ⇒ but different actions, tasks.

* FEATURES OF JAVA PROGRAMING:

Application software can be built

JAVA CORE

Multithreading
Interfaces
I/O handler
Java packages
Exception handling.

Interface design. GUI

JAVA APPLET

Java multimedia
Java swing
windows toolkit
Java Script.

JAVA INTERNET

Networking
DB ↑ Connectivity.
JDBC

JAVA PROGRAMING STEPS
#LECTURE-2

import java.lang.*;
C, JAVA are Case sensitive.

ASPECTS	C	JAVA
Paradigms	Procedural	OO
Platforms	Dependant	Independant
Datatypes: unions, str.	✓	α
Pre-processor dir.	✓	α
Header files	✓	α
storage class	✓	α
Inheritance	α	✓
Pointers	✓	α
Code translation.	COMPILED	INTERPRETED
Multi threading ↓Interface.	α	✓
Exception handling	α	✓
DB Connectivity	α	✓

• Java extension.

Editor of any kind — emac, gedit, MS, txt editor etc.

TRANSLATION ⇒ JAVAC "Java" TO "class"

↳ Java compiler.

JAVA "" without class

What does platform independent mean?

* Compiled file can be understood by any machines. (.class)

C → functional

C++ → object

JAVA → object.

#C EXECUTION

Source code .c

↓ Preprocessor

↓ Compiler

↳ ASSEMBLY CODE

Assembler ↲

libraries → linker ← ↳ OBJECT FILE .o

↓

EXECUTABLE

#JAVA

@ compiletime

.JAVA SOURCE FILE → (compiler) → .CLASS BYTECODE

↓

@ runtime

IIT compiler (JVM kg.) → Native code

Bytecode targets a hypothetical machine irrespective of architecture

↳ IIT have all types of architecture. and convert this bytecode.

C++ ⇒ Large software development

JAVA ⇒ + Communication / Internet application software.

FEAT:	C++	Vs	JAVA
Data abs + encapsulation	✓		✓
Polymorphism	✓		✓
Binding (static + dynamic)	✓		✓
Inheritance SINGLE	✓		✓
Inheritance MULTIPLE	✓		×
operator overloading	✓		×
Template class	✓		×
Global var	✓		×
Header var.	✓		×
Pointer	✓		×
Interface + Package	×		✓
API	×		✓

⇑

NOT PLATFORM INDEPENDENT.

JAVA TOOLS AND RESOURCES
#LECTURE-3

JDK - Java development kit.

Javac, Java
Javadoc – gen documentation in HTML.
Appletviewer - Java interpreter to execute Java applets
Jbd - Java debugger
javah - Create interface btw Java & C routines.

"Javatpoint website"

API ⇒ 9 Packages. → rel to GUI, designing.

Java.applet ⇒ for applet programming
java.awt ⇒ ABSTRACT WINDOW TOOLKIT.
java.io ⇒ i/o
java.lang ⇒ ~~object~~ It handles object, thread, exception, math, float, int, system.

java.net ⇒ network programming.
java.util ⇒ miscellaneous classes
VECTORS, STACK, LIST, DATE, HASH, DICTIONARY.

java.awing ⇒ GUI dev, extension version of awt
java.sql ⇒ JDBC.

* RICH SUBSET OF LANG...

Built in datatypes
System
Boolean
Assignment
Arrays
Flowcontrol
Primitive numeric types
String
Parsing
Math library.

There are two types of data types
→ PRIMITIVE DT
→ REFERENCE DT

ARRAY ⇒ declaration of array — int [] x
Allocate memory for it. — x = new int []
load val in array.

int [] x = {1, 2, 3, 4};
int x[][] = new int [3][4];

* We can create a variable sized 2D-array.

JAVA APPLET PROGRAMMING
#LECTURE-5

Applet is like a window/display.
GUI, web.
Applet components ⇒ buttons, etc.
@ web ⇒

```
<body>
    <applet code = "Hello.class" width = " " height = "
</body>  </applet>.
```

while running the html doc.
appletviewer index.html.

```
import java.applet.Applet;
import java.awt.Graphics;

Public class helloworld extend Applet {
    Public void paint(Graphics g){
        g.drawstring(" ", 150, 150);
    }
}
```

Applet is depricated....

BASIC METHODS IN APPLET

init() → initialis/ pass I/o.
start() → starts applet.
stop() → stops applet.
paint(Graphics g) → to draw something within applet.
destroy() → rem. apple from memory.

"NOT INCLUDED IN SYLLABUS"