

JAVA

Developed at Sun Microsystems which was purchased by Oracle in 2010.

- General purpose & powerful programming language.
- Used for developing software that can run on mobile, desktop & servers.
- Machine independent

The syntax & semantics of Java

- To write English we should follow some rules (Grammars)
 - Also, to write Java we should follow some rules → Syntax & semantics
- Here are playing → Syntax error (Grammar)

He is hello & lec → semantic error (Meaning)

API

Application programming interface

Also known as library
Contains predefined Java code that we can use to develop Java programs.
→ Faster & easier development process & no need to write everything from scratch.

Java has 3 editions

- Java Standard Edition (SE): develop applications that run on desktop
- Java Enterprise Edition (EE): develop server-side applications.
- Java micro Edition (ME): develop applications for mobile devices.

JDK (Java development kit)

- Set of programs that enable us to develop our programs.
- Contains JRE (Java Runtime Environment) that is used to run our programs.
- JRE & JDK contain JVM (Java virtual machine)
- JVM executes our Java programs on diff. machines

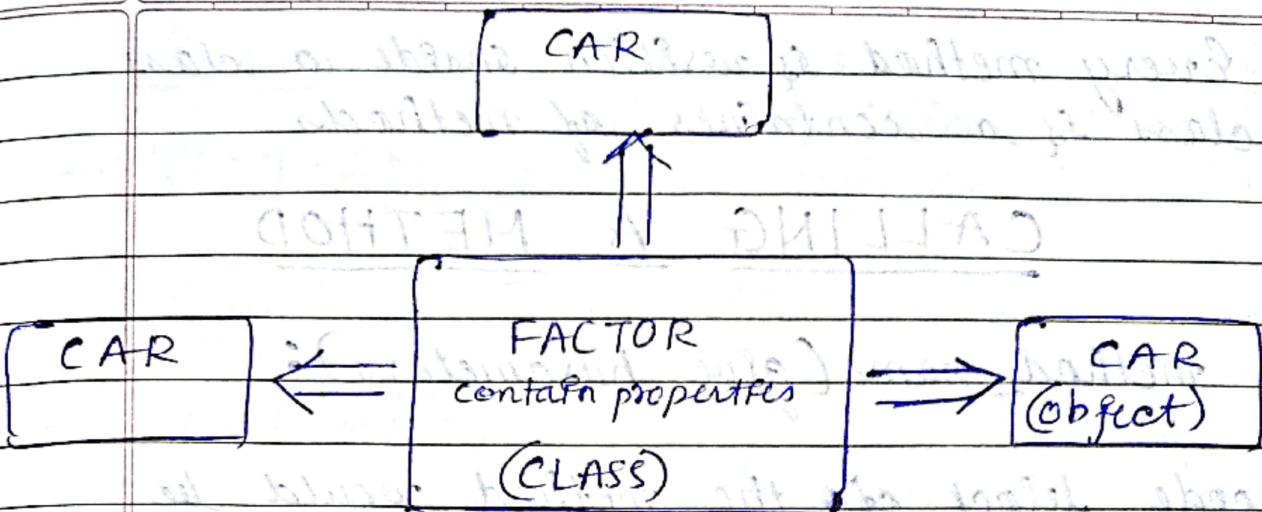
IDE (Integrated development environment)

A program that allows us to:

- Write source code → Debug tools & find errors
- Compile machine code of errors
- Build files that can → Execute our be executed by JVM.

CLASS → A template to create objects

OBJECT → An instance of a class



CLASS STRUCTURE

class class_name { } first one will be address

METHOD → Group of instructions to do a specific task

E.g.: add 2 nos.

say hi to user

We have a special method called 'main'.

METHOD STRUCTURE

Each method consists of 4 parts

return_type method_name(parameters) { }

code block

4

Note: Every method is written inside a class
→ A class is a container of methods

CALLING A METHOD

method_name(give parameters);

→ The code block of the method would be executed.

Note: The main method is automatically called when we run our Java program.

→ It is the first method that is called

→ It is the starting point of execution of our program

ACCESS MODIFIERS

Specify how to access classes & methods

- Public
 - Private
 - Protected
 - Default
- These are keywords for modifying access.

NAMING CONVENTIONS

How to write names in programming:

Pascal case convention:

→ ThisIsAName first letter of each word is uppercase.

Camel case convention:

→ thisIsAName only first letter of 1st word is small thereafter each new word has the first letter capital.

Snake case convention:

thes_is_a_name Put underscore betw. each word.

JAVA PROGRAM STRUCTURE

public class Main {

 public static void main(String[] args) {

 System.out.println("Hello Java");

→ Every Java program contains at least one class.

→ Pascal case is used with classes.

→ Camel case is used with methods.

→ A method exists inside a class.

→ main() is the starting point of execution of

our program.

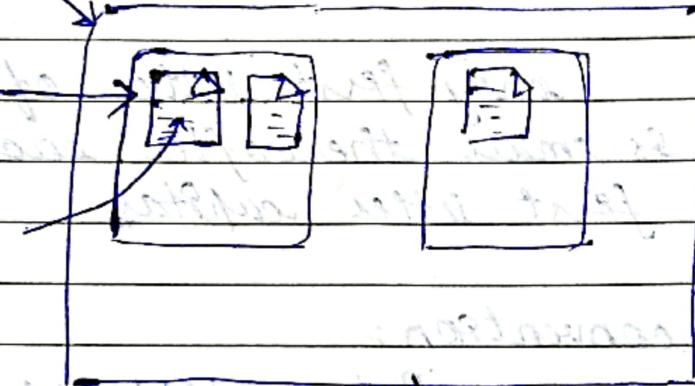
PACKAGE

A container for classes

package

class

method



STRINGS

Strings in Java should be put in " double quotes

E.g. of strings

"hello" FORTUNE MARTIN AVAT
"1"

"123"

"1@#123**"

CALLING PRINTLN()

Displays its parameter on the console window.

System.out.println("Hello"); output hello

System.out.println("123"); output 123

'ln' goes to break for a new line.

CALLING PRINT()

System.out.print("Hello"); → hello123
 System.out.print("123"); → 123

→ No line break after two methods.

S → beginning (E-E) without line break.

What is System.out?

- out is an object of the 'PrintStream' class.
- out has the print() & println() methods.
- Use '.' to access print()/println() of out.
- out refers to the standard output device(screen).
- System is a class (Pascal case).
- out is inside System(field).
- Use '.' to access out of System.
- System.out.println().

NUMBERS

Numbers in Java are just like in mathematics.

E.g.

0, -19, 129, -0.5, 90.81

ARITHMETIC OPERATORS

Addition, subtraction, division, multiplication.

Addition operator: +

Subtraction -

Multiplication *

Division /

`System.out.println(5 + 2);` → 7

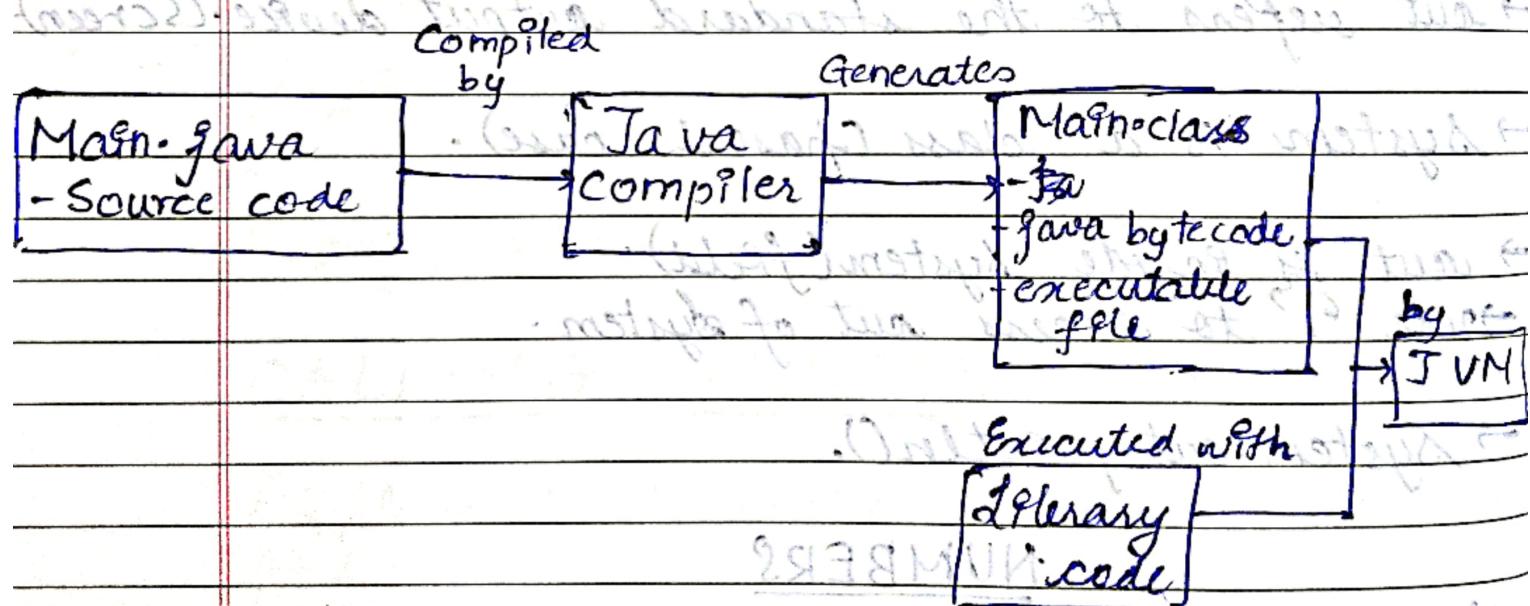
`System.out.println(3 - 5);` → -2

`System.out.println(6 * 3);` → 18

`System.out.println(4 / 2);` → 2

The value will be calculated & printed.

Save Java files as .java & name of the file should be same as name of the class.



public class Welcome

public static void main(String[] args){}

`System.out.println("Welcome to Java");`

O/P : Welcome to Java

COMMENTS

// this is a single line comment

/* this is a multi-line comment */

Project name: LearnJava

Project location: C:\Users\A puro\Idea project\

Base package: com.Apruad

package com.Apruad;

public class Main {

 public static void main(String[] args) {

 // write your code here

 System.out.println("Hello!!");

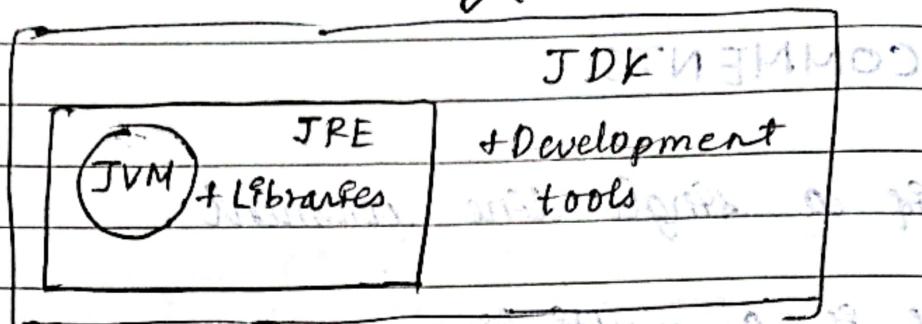
 }

}

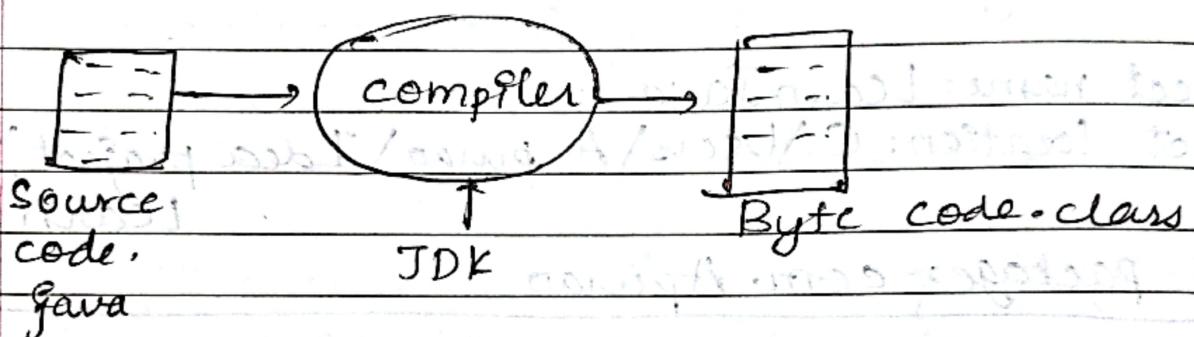
O/P : hello!!

Name of class can be anything

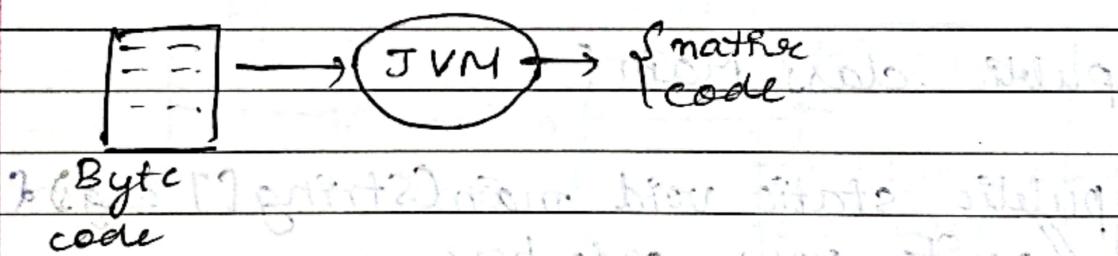
How is code running?



COMPILATION



EXECUTION



Java Class

1. Function

```
void main() {
```

^y

2. class

```
class Main{
```

```
    void main(){
```

^y

Output in Java

```
System.out.print("Hello World");
```

① print

System.out.print("...");

② print

② println

③ "\n"

Data Types

Primitive

byte

short

char

boolean

int

long

float

double

Non-Primitive

String

array

Class

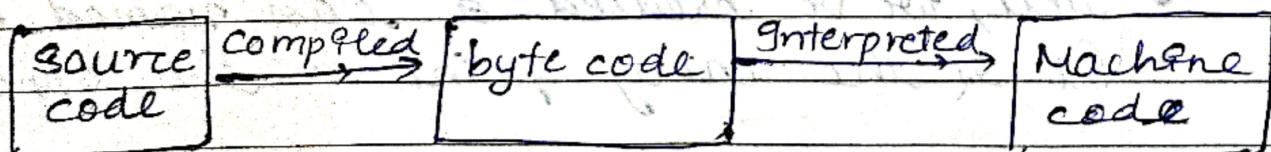
object

Interface

Java was developed by sun microsystems
of USA in 1991.
It was originally called Oak by James Gosling.

JAVA = Purely Object Oriented

Java is compiled into the byte code and
then it is interpreted as machine code.



First Java Program

```
package com.company;  
public class Main {  
    public static void main (String [] args) {  
        System.out.println ("Hello World");  
    }  
}
```

Basic Structure

1) package com.company:

- Packages are used to group the related classes
- The "Package" keyword is used to create packages in Java
- Here, com.company is the name of our package

2) public class Main:

- In Java, every program must contain a class
- The filename and name of the class should be the same
- Here, we've created a class named "Main". It is the entry point to the application.

- 3) `public static void main(String[] args){...}`
- This is the main method of our Java program
 - Every Java program must contain the main method

4) `System.out.println("Hello World");`

- The above code is used to display the output on screen.
 - Anything passed in the quoted commas is printed on screen as plain text
- For classes we use Pascal Convention.
- For functions and variables we use camel case convention.

Documentation section: → Suggested

Package statement: → Optional

Import statements: → Optional

Interface statement → Optional

Class Definitions → Optional

Main method class

Main method defⁿ → Essential

Ch.1 Variables & Data Types

Date _____

Page _____

Just like we have some rules that we follow to speak English (the grammar), we have some rules to follow while writing a Java program. The set of these rules is called Syntax.

Variables

A variable is a container that stores a value. This value can be changed during the execution of program.

Data type $\text{int number} = 8;$ value it stores
variable name

Rules of declaring a variable name.

- 1) Must not begin with a digit
Eg. 1arry is invalid
- 2) Name is case sensitive
E.g. Adwart and adwart are different
- 3) Should not be a keyword (like void)
- 4) White space not allowed
E.g. Adwart is a good boy
- 5) Can contain alphabets, and \$ characters,
- characters & digits if other conditions
are met

Data Types

1) Primitive Data Types (Intrinsic)

2) Non-Primitive Data Types (Derived)

PRIMITIVE DATA TYPES

Java is statically typed i.e. variables must be declared before use.

Range of values: -2^{n-1} to $2^{n-1} - 1$

There are 8 primitive data types in Java:

- 1) byte →
 - Values range from -128 to 127
 - (stores integer)
 - Takes 1 byte $-(2^7)/2$ to $(2^7)/2 - 1$
 - Default value is 0
- 2) short →
 - Values range from $-(2^{16})/2$ to $(2^{16})/2 - 1$
 - Takes 2 bytes
 - Default value is zero
- 3) int →
 - Values ranges from $-(2^{32})/2$ to $(2^{32})/2 - 1$
 - Takes 4 bytes
 - Default value is zero
- 4) float →
 - Takes 4 bytes
 - Default value is 0.0f

5) long → • value ranges from

$$-(2^{64})/2 \text{ to } (2^{64})/2 - 1$$

• Takes 8 bytes

• Default value is zero

6) double → • Takes 8 bytes

• Default value is 0.0d

7) char → • Value ranges from 0 to
 $65535(2^{16}-1)$

• Takes 2 bytes → because it supports
unicode

• Default value is '\u0000'

8) too boolean → • Value can be true or
false

• Size depends on JVM

• Default value is false

How to choose data types for our variables

Primitive Data Types

Integral (int)

byte

short

int long(L)

Floating

float double
(For f) (For d)

Character

('')

Boolean
(true/false)

In order to choose the data types we first need to find the type of data we want to store. After that we need to analyze the min. & max. values we might use.

Literals

A constant value which can be assigned to the variable is called as a literal.

10 → Integer Literal

10.1 → Float Literal

10.1 → double Literal (default type for decimal)

'A' → character Literal

true → boolean Literal

"Harry" → String Literal

For String

```
String str = "Adwait";
```

Keywords:

Words which are reserved and used by the Java compiler. They cannot be used as identifiers.

→ Go to docs.oracle.com for a comprehensive list.

Reading data from keyboard

In order to read data from the keyboard, Java has a new Scanner class. Scanner class has a lot of methods to read the data from keyboard.

```
Scanner S = new Scanner(System.in);
```

Read from keyboard

```
int a = S.nextInt();
```

↳ Method to read from

Keyboard (Integer in this case)

```
float a = sc.nextFloat();
```

For strings

```
String str = sc.nextLine();
```

↳ Does print string

"HelloA" with spaces

```
String str = sc.nextLine();
```

↳ Prints Read

String with spaces

Scanner class of java.util package is used to take input from the user's keyboard. The Scanner class has many methods for taking input from the user depending on

the type of input. To use any of the methods of the Scanner class, first we need to create an object of the Scanner class as shown below.

```
import java.util.Scanner;
```

// Importing the scanner class

```
Scanner sc = new Scanner(System.in);
```

// Creating an object named "sc" of the Scanner class

Q 2 Write a program to calculate percentage of a given student in exam. His marks of 3 subjects must be taken.

```
package com.company.main;
```

```
import java.util.Scanner;
```

```
public class Percentage
```

```
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter the name of your ward: ");
```

```
        String str = sc.nextLine();
```

```
        System.out.print ("Percentage of: ");
```

```
        System.out.println (str);
```

```
        System.out.println ("Enter the marks of sub1");  
        float a = sc.nextFloat();
```

same for sub2 & sub3

float percentage = (a + b + c) / 3;

System.out.print("The percentage is: ");
System.out.println(percentage);

3

O/p

Enter the name of your ward
Adwait

Percentage of: Adwait

Enter marks of sub1

45

67.90

99.99

The percentage is: 70.96333

Access Modifiers

PUBLIC → The access level is

everywhere

→ Inside a class outside a class

→ Inside the package

→ Outside the package

PRIVATE - The access level is only inside the class.

STATIC - You can access fields/methods using the class name.

Practice the set.

Q1 Write a Java program which asks the user to enter his/her name & greets them with "Hello <name>, have a good day" text.

package com.company;

public class Adwait

```
public static void main(String[] args) {
    System.out.println("What is your name");
    Scanner sc = new Scanner(System.in);
    String name = sc.next();
    System.out.println("Hello " + name + " have a
                       good day");
```

O/P : what is your name

Adwait

Hello adwait have a good day

Q2 Program to check integer or not

package com.company;

public class prob_2

psum (String [] args)

System.out.println("Enter a number");

Scanner sc = new Scanner(System.in);

System.out.println(sc.hasNextInt());

y

3

O/P

Enter your number

1

True

Enter your number

Ad

False