#4 First Java Program.

* Structure of Java File :-
  "Source code that we write will be saved using extension [.Java] file."

- Every thing written in .Java file must be in classes or we can say that every file having .Java extension is a class.
- A class with same name as file name must be present in .Java file.

First Alphabet of class name can be in upper case. it is the naming Convention of a class name. (It's not Compulsory) But it is good practice.

* Converting .Java to .class :→

* Using Javac Compiler We can Convert .Java file to .class file.
  Command to Convert it is:
  "Javac & .Java file name"

* Above Command creat a .class file (Main.class) Which Contains Bytecode.

* Running the Program:- By using java & file name we can run java Program.
  Command > Java (Main) → file name.

* "Hello World" Program:-

```
Public class Main {
    Public static void main (String [] args){
        System.out.Println ("Hello world");
    }
}
```

(1st line)

① Public :- it is the access modifier which allows to access the class from anywhere.

② class :- It is a name group of properties & funct

③ Main :- it is just name of class as name of file.

④ Public (2nd line) :- It is used to allow the program to use v function from anywhere.
Main

⑤ static :- It is the keyword which helps the main method to run without using objects.

⑥ void :- It is a keyword used when we do not want to return anything from method / function

⑦ main :- It is name of method.

⑧ string [] args :- It is command line argument of string type array.

⑨ System :- It is the final class defined in Java.lang. Package.

(10) out :- It is variable of printstream type which is public & static member of field of system class.

(11) Println :- It is the method of Printstream class., it prints the arguments passed to it & adds a new line.

* instead of println, if we remove "ln" only print can take, it will print next input on same line. not on a new line.

* What is Package ?

→ Package is just a folder in which java file lies.

→ it is used to provide some rules & stuff to our programs.

* Primitive Data Types :-

- Primitive data types are those data types which is not breakable.

Ex.

String is not primitive data type so we can break into char.

i.e String "Omkar" can be divided into

'O', 'm' 'k' 'a' 'r'

we can't break char, int. etc.

Data Types	Description	Example
int	used to store numeric digits.	int i = 26;
char	used to store character.	char b = 'A';
float	used to store floating point no.	float f = 98.67f;
double	used to store larger decimal no.	double d = 45678.567
long	used to store numeric digits which are not able to store in int	long l = 154326789
boolean	only stores 't' values, i.e. true or false	boolean b = false;

* Literals :- it is synthetic representation of boolean, character, string, & numeric data.

Ex. - int a = 10;

Here 10 is called literal.

* Identifiers :- name of variables, methods, class, packages, etc are known as identifires.

Ex. int a = 10;

↳ Identifier.

* Comments in Java :-

Comments are something which is written in source code but ignored by Compiler.

* Two types of Comments :-

① Single line Comment	② Multiline Comments.
→ used to Comment down a single line.	→ used to Comment down multiple lines.
(// is used for it)	(/* */ is used for i

* Inputs in Java :-

We have Scanner class avaidliable in Java.Ut Package to take input.

To use this Class we have to ;

① import Java.Util Package in our file.
② Creat object of scanner class.
③ Use that object to take input from the keyboard.

Syntax :-

```
import Java.util.Scanner;
Public class main {
    Psvm (shortcut of line) {
        Scanner input = new Scanner (sys
    }
}
```

★ Type Conversion:-

when one type of a data is assigned to the another type the automatic type Conversion will take place uned some Condition.

★ Conditions:-

① Two types should be Compatible.

② Destination type should be greater than source type.

★ Type Casting:-

When we Convert one type of a data to another type is known as type casting.

Ex. int num = (int) (67.564f);

★ Ex. of if statement:-

statenit inside if statement only executes when Condition given is true:

```
Public class Main {
    PSVM {
        int a = 10;
        if (a == 10) {
            System.out.println("Hello ~~output~~");
        }
    }
}
```

Output => Hello

* Example of While loop :→

Statement in While loop run till Condition in While loop become false.

```
Public class While loop {
    psvm , ——— , — {
        int count = 1 ;
        while ( count != 5 ) {
            sout ("count") ;        ( sout = Syst.out.println)
            count++
}  }  }
```

* Output = Count
count
count
Count.

* Ex. of for loop :-

```
Public class for loop {
    p s v m {
    for (int count = 1 ; count != 5 ; count++) { {
        sout (count);
    }
}  }
```

Output => 1
2
3
4