First Java program

Lecture - 3

Few Commands :-

open Command prompt & write

Two commands.
- javac -version
- java -version.

to check we have correct JDK version installed on our laptop. if not, install it. install it

So now we will write our first Java program in any Text editor. Let's open Note pad and write.

first create a directory (folder).

Command md foldername. (or) mkdir foldername

keyword

```
Class firstprogram {
    public static void main(String[] args){
        System.out.println("Hello welcome to
                            ManikLearns.. blog");
    }
}
```

class name.

Save it as firstprogram.java.

Why class name & filename is same?

* It need not to be same for every Java prgm. Here we can have a different name to save this file.
* But if the class is public then it is necessary to have both class name & file name same.

* And generally we declare class public to make it accessible everywhere. If we don't declare class public to make it accessible everywhere if we don't declare it public then the class would be package-private means it can not be used in another package.

=> packages helps group related classes making it easier to manage & maintain code, avoid naming conflicts and control access

NOTE:- Conceptually, it is similar to folders on your laptop. So this same name. Convention was set by Sun, so that everyone who code in Java will have a consistant way of naming files.

Why haven't included any header @ package here like we include in C & C++ (#include <stdio.h> & #include<iostream>) to prgm work?

The Java.lang package is automatically imported by Java compiler for every java prgm so no need to import it explicitly. If you want you can write:

```
import java.lang.*;
```

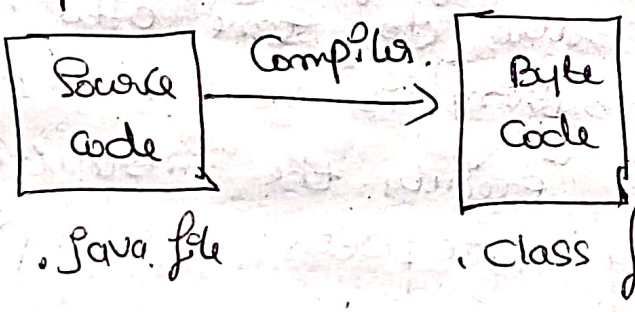
-> But if we want to use classes from other packages, then we have to import those explicitly.

→ java.lang package is one of the core package in java & it contains fundamental classes that are essential for java progming.
Eg:- System, String, Object, Thread, Math (to perform basic mathematical functions), exception etc.

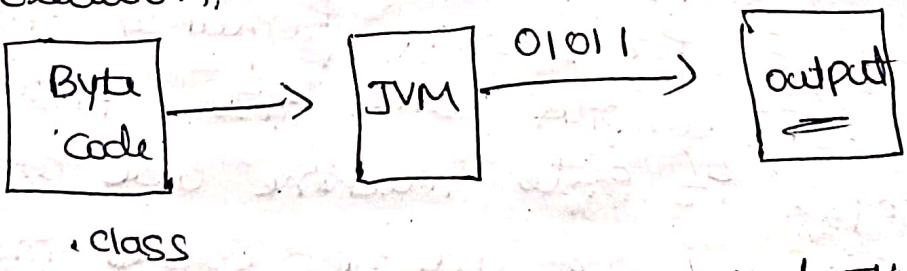
Understanding the Compilation and execution flow of the prgm

This is two step process.
① Compilation.
② Execution.

① Compilation.



② Execution,



→ Bytecode is an instruction set of JVM & each instruction is represented by 8 bits (one byte). Hence it is known as bytecode It is compact form of data.

How JVM converts bytecode / How JVM works?

→ JVM is abstract computing machine having instruction set (bytecode) & manipulates memory areas at run time.

→ JVM knows nothing of Java programing language but it knows the class file format. A class file contains Java virtual machine instruction and a symbol table and some other ancillary information.

when we execute the command then JVM would be created memory and it also has some JVM memory from OS

→ Now class loader loads the .class file into memory (JVM memory). Now the bytecode would be verified by bytecode verifier to ensure that the bytecode adheres to the rules of the Java language & doesn't violate access rights and also do some other checking on bytecode if everything is ok in the bytecode, execution engine executes the instruction. with the help of interpreter JIT compiler. Interpreter interprets the code line by line & execute & for some frequency called methods JIT compiler comes into picture & may compile it into machine code and then this machine code is cached in memory. Subsequent call to that method will execute the native code instead of interpreting the bytecode again. Significantly speeding up execution.

And then you get the output

understanding the Structure of first pgm

```
class FirstProgram {
}
```

FirstProgram → name of the class

class → keyword / reserved word.

(while naming the class we should follow upper camel case (or) pascal case convention)

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

main method.

→ main methode:- PS public that everyone can access it if we don't make it public JVM will not be able to access it, pgm will be compiled correctly. but will you run it will give error. "main method not found". So we have to make it public to make it accessible for JVM.

→ static:- pgm execution starts from main() method. declaring it static allows the JVM to invoke the main method without creating an instance of the class containing it.

* If main were not static, the JVM would need to creat an object of the class before invoking the method, which would lead to complexities.

* The main method is directly called by the JVM when pgm starts. The static main method is a convention (or) standard entry point in Java as well as in some other pgming language, making

It easier for the JVM & compiler to recognize the program starting point.

->void is just a return type, void. means there is no return value main is the name of this method.

* If we don't declare main method as static. program will be compiled correctly but it will not run. It will give error "main method is not static".

* ~~Static~~ String[] args :- There are arguments to main method, this method must be given an array of Strings and the array name is. args (we can take any array name other the args if we want).

Eg:- String[] a is also fine

It is a way to pass command line arguments [we will discuss this thing in later.]

The String[] args parameter in the main method is part of the standard Java method signature for the program entry point. without it JVM won't recognize the main method & may not be able to start the prgm.

without String[] args the prgm will compile successfully but will not run because of the violation of the conventional signature expected by JVM.

→ System.out.println("Hello welcome to monikalea blog");

This is a way to print output to standard output in Java

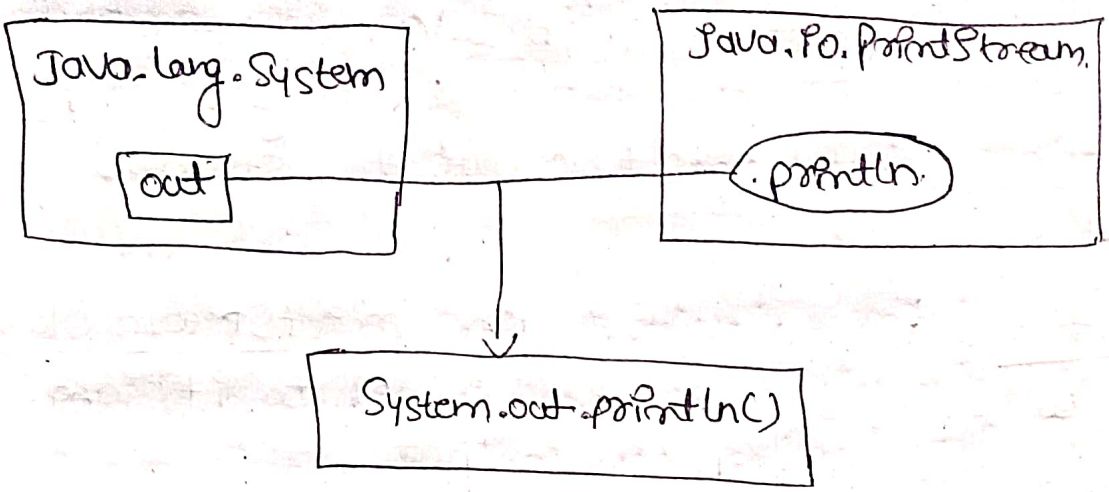
→ println() is a method in printStream class used to print any argument passed to it and adds a new line to the output.

out is an instance/object of printStream class and out is a static member of System. class out member/field represents the standard output stream (usually the Console) to which text and data can be printed.

System is a final class in the java.lang package. Since java.lang package is imported automatically by the JVM we don't need to explicitly import System. System class provides various fields & methods for standard input, output & error streams & much more.

→ So Basically System.out refers to the standard output stream typically connected to the Console. we can redirect this output to a file or another output destination if necessary.

→ we can use print() method rather than println(). print() also prints the arguments passed to it but it doesn't add a new line after the output.



practice questions

① write a prgm to print your name, age, and hobby on seperate lines

```
public class MyHobby {
  public static void main(String[]args ){
  System.out.println ("my name is Manika");
  System.out.println("I am 20 years old");
  System.out println(" my hobby is listening
                      to music");
  }
}
```

② write a Java prgm to print the following pattern

i.
```
*
* *
* * *
```

ii.
```
  *
 * *
* * *
```

```
public class pattern{
  public static void main(String[] args){
System.out.println("*");
System.out.println("* *");
System.out.println("* * *");
  } }
```

```
pp. public class pattern{
    public static void main(String[] args){
        System.out.println("    *");
        System.out.println("  * *");
        System.out.println("* * *");
    }
}
```

③ write a prgm to print sum of integer (10+15) using System.out.print() & System.out.println()

```
public class Addition{
    public static void main(String[] args){
        System.out.println("The sum of 10 and 15 is");
        System.out.println(10+15);
        System.out.print("The sum of 10 and 15 is");
        System.out.print(10+15);
    }
}
```

④ what would be the output of the following code:

```
Class practiceprgm{
    public static void main(String[] args){
        System.out.print(10+5);
        System.out.println("10");
        System.out.println(12.156);
        System.out.println("10+15");
        System.out.println('a');
    }
}
```

O/p:-

```
15 10
12.156
10+15
a
```

Introduction to JShell

→ JShell is an interactive tool for learning the Java programing language and prototyping Java code.

→ JShell is a Read - Evaluate - print loop (REPL) which evaluates declaration statements and expressions as they are entered and immediately show the results. Seen

→ This tool is run from the Command line. JShell was introduced in Java 9

Starting and Stopping JShell.

→ To start JShell, enter the jshell command on the command line.

NOTE:- JDK 9 @@ higher version must be install on your system.

To exit JShell, enter /exit.

To start Jshell in verbose mode, use the -v option

jshell -v.

Code snippets

* JShell accepts Java statements, methods, variables & class definition, import & expressions. These pieces of Java code is known as snippets

Eg:- jshell > int a = 5;

a ==> 5.

| created variable a : int.

(In verbose mode the description also shows

NOTE:- Semicolons are automatically added to the end of a complete snippet if not entered.

* when an expression is entered that doesn't have a named variable, a scratch variable is created so that its value can be referred later

Eg:- Jshell > 3+5
$1 ==> 8
I created scratch variable $1 : Int

-> Jshell > void display(){
...> System.out.print("Hello");
...>}
Created method display().
Jshell > display();
Hello.

there are many Jshell commands used to control the environment and display information.

Eg:- /var -> gives information about current variables
/methods -> gives info about current methods.
/list -> gives a list of entered snippets.

NOTE:- Jshell has a default startup script that is silently & automatically executed before Jshell starts, so that we can get to work quickly, to show all the entries from startup.

use case. /list -all (or) /list -start
Command /help → list all the JShell Commands.

So JShell is a REPL for Java. which means:-

i. It reads the Command. (or) Code Segment we type.
ii. it evaluates and execute the code
iii. It prints the result/output without making developer write code to output the result
iv. And then it loops right back for more in

NOTE:- JShell doesn't replace an IDE

Few Definitions.

Statement :- It is a complete Command to executed. It is an instruction for Comput to do Something

Eg:- System.out.print (3+5);

Expression :- expressions are part of Statem An expression is a Combination of Variables, values & operands that produce result (or) value when evaluated..

Eg:- 3+5
(or) (a+5)*3

expression always return a value and ca be part of Statements.

int a= 5+3;] => Statement

int a;
x =10; } => Statements

↳ 10 is an expression.

So expressions produce a value when evaluated Statements perform an action and may ⊛ may not contain expressions.

Can we have more than one class in a java file?

yes you can but only one class can be public & the name of public class & name of java file must be same.

But it is not compulsory to have a public class all classes can be without public specifie

Suppose we have 3 classes in one java file. So after compilation three .class files would be generated. So in this case we are not able to identify which class contains main method (entry point) for JVM to execute prgm;

One solution is check the source code & find out which class has main method. but this is tedious work for any one. So we are forced to follow some convention rules set by Sun, just for better organization of the code. So its recommended to have one public class per java file but if you have multip classes in single java file then make one class public which is having main method & save the file with some name as public class name. So that while running its easy to Identify which class has main & to be executed.

Better to have nested classes if you need multiple classes in java file. It is not compulsory that the public class should have main method.