emailid :: nitin@ineuron.ai
linkedinid:: Nitin M(https://www.linkedin.com/in/nitin-m-110169136)

Topic(13 sessions)
==========
=> Need of Compiler vs Interpreter
=> Usage of identifiers,reserve words
=> Need of datatype in Java
=> Operators and Control statements
=> Arrays
=> Strings
=> JVM Architecture
=> OOps
=> Exceptionhandling
=> MultiThreading
=> Collections
=> JDK8 features
=> JDBC,Servlet,JSP
=> Hibernate
=> SpringBasics,SpringBoot and Microservices

Note:: DSA is requried only for product based companies(FANG,MANG,...highest package[1.8cr])

Compiler -> It is a software which takes sourcecode(HLL) as the input and generates MLL code as the ouput
To convert the HLL code to MLL code compiler will scan the HLL code only once.

Interpreter =>It is a software which takes sourcecode(HLL) as the input and generates MLL code as the ouput
To convert the HLL code to MLL code interpreter will scan the HLL code mulitple times(depends on the instrcutions).

Performance measurement of Compiler vs Interpreter
==========
=> Compiler will speed up the process ,where as intrpretter will slow down the process.
=> Compiler in one Scan will identify all the problems in the code(if found),where as interpretter will do scanning line by line so
it takes more time for identifying the problem.

How does java program runs?
=> java program to run we need to set up the environment in our machines
=> To set up the environment we need to install JDK software to our machines.
=> JDK stands for Java Development kit, it provides libraries and the requried files to run our java programs
=> JDK :: JRE + JVM
=> JRE :: Java RunTimeEnvironment, It provides suitable environment to run our java program.
=> JVM :: Java Virtual Machine, It is responsible to run our java programs on the basis of MultiThreading.

Program Execution
==========
=> java programs will be first compiled,to compile java program we need java compiler and java compiler will be installed when we install
jdk software.

```
 => jdk software installation location :: c:\programfiles\jdk....
 => All the commands required for java developer to run his program would be
present inside <java_home>\bin folder
 => bin folder
      |=> javac
      |=> java
      |=> javap
      |=> javadoc
      |=> jar

 => javaprogram would be first compiled(javac filename.java)
 => If the compilation is succesfull, it would generate .class file
 => These .class files will be used by jvm during the execution

 => .class file generated will have instructions in bytecode(neither HLL nor MLL).
 => bytecodes will be taken by JVM and it will be loaded inside JRE, then the
execution begins.


Java programs => java compilation + Execution
                   (javac)          (java-> JIT)

Rule followed while writing a java program
==========================================
 => In single source code,we can write only one class under public category
 => Which ever class is under public category,that class should hold main() and
that classname should be saved as the filename.
 => Main method will be used by JVM during the execution, so java program expects
the programmer to write main() in the following style
            public static void main(String[] args)



What is an identifier?
 identifer is a name in java program.
 identifer can be classname,methodname,variablename,labelname.

eg::
class Demo
{
      public static void main(String[] args)
      {
            int x = 10;
      }
}

Demo: classname
main: methodName
String: className
args  : variablename
x : variableName

What are the rules followed to define a java identifier?
  => The only character allowed for java identifiers is
  => a to z, A to Z,0 to 9,$,_

Can identifier start with digits? No
Is there a restriction on the lenght of identifiers? No
We can't use reserve words as an identifer.
```

We can use inbuilt classnames and variablenames as identifiers,but it is not a good practise to use.

```
eg::
public class Sample
{
	public static void main(String[] args)
	{
		int String = 20;
		System.out.println(String);//20

		int Runnable= 25;
		System.out.println(Runnable);//25
	}
}

eg::
public class Sample
{
	public static void main(String[] args)
	{
		int System = 20;
		java.lang.System.out.println(System);
	}
}

eg::
public class Sample
{
	public static void main(String[] args)
	{
		int Thread = 20;
		System.out.println(new Thread());
	}
}

eg::
public class Sample
{
	public static void main(String[] args)
	{
		int String = 10;
		System.out.println(args.length);
	}
}
```

Note:: JVM preference while loading the .class file

1. Search in the current working directory from where the program would start.
2. Search in the rt.jar(inbuilt jar with all the .class files) file also with the specified packages
3. search in the ext folder of jre environment.
4. search in the classpath supplied jar file

```
eg::
public class String
{
	public static void main(java.lang.String[] args)
```

```
	{
		System.out.println("Hello World!");
	}
}

Flow of the code
================
D:\PPT program for Java>javac String.java
		|-> String.class

D:\PPT program for Java>java String
			|=> JVM
		String.main(new String[]{});

What is a ReserverWord?
  They are builtin words associated with special meaning.

How many reserve words in java ?
  Reserve words -> 53

Keyword -> 50
	a. used keywords(48)
	b. unused keywords(2)
		a. goto
		b. const
Reserved literals -> 3
	a. null
	b. true
	c. false


Reserve words for datatypes(8)
	byte,short,int,long,float,double,boolean,char

Reserve words for flow control(11)
	if,else,switch,case,default,for,do,while,break,continue,return

Reserve words for modifiers(11)

public,private,protected,static,synchronized,strictfp,final,abstract,native,transie
nt,volatile

Reserve words for ExceptionHandling(6)
	try,catch,finally,throw,throws,assert(1.4v)

ClassRelated keywords(7)
	enum,class,package,interface,extends,import,implements

ObjectRelated keywords(4)
	new,instanceof,super,this

keywords for returntype
	void



For the code below, what should be the name of java file?
1.
public class HelloWorld {
```

```
    public static void main(String [] args) {
        System.out.println("Hello World!");
    }
}
```

A. Hello.java
B. World.java
C. HelloWorld.java
D. helloworld.java

Answer:: C

2.
Does below code compile successfully?

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

A. yes
B. no

Answer: A

3.
What is the signature of special main method?
A. public static void main(String args)
B. public static void main(String[] a)
C. public static void main()
D. private static void main(String[] args)

Answer::B

Command line arguments
======================
=> The arguments which are passed from the command line to our java program is called "CommandLine-Argument".
=> Java programs execution starts from main()
=> If we want to pass something to main() while running the java program from the command prompt we need to go for "CommandLine Arguments".
=> These command line arguments are taken by jvm and jvm will refer to the filename supplied by the user.
=> Using the filename and the arguments supplied, jvm will call the main method by creating an array of String type with the supplied argument.
=> if the arguments are not supplied then the array size would be zero,if it is supplied then that data would be a part of String array.

```
java FileName [Arguments]
    |
    |JVM
    |
  FileName.main(new String[]{["Arguments"]})
```

4.
What will be the result of compiling and executing Test class?
  java Test good morning everyone

```
private class Test{
   public static void main(String args[]) {
       System.out.println(args[1]);
   }
}
```
A. compilation error
B. good
C. morning
D. everyone

Answer : A[class is under private,so it is an CompileTimeError]

5.
For the class Test, which options, if used to replace /*INSERT*/, will print "Hurrah! I passed..." on to the console? Select 2 options.

```
public class Test {
    /*INSERT*/ {
        System.out.println("Hurrah! I passed...");
    }
}
```

A. static public void main(String[] args)
B. public static void main(String[] a)
C. static public void Main(String[] args)
D. public void main(String[] args)
E. protected static void main(String[] args)
F. public void static main(String[] args)

Rule: we can interchange access modifier before the return type.
      array variable name can be anything.

```
public static void main(String[] args)
static public void main(String[] a)
```

Answer:: A,B