**coreJAVA || Day2 ||20 july 2022**

**==========================================================**

1) Explore JConsole (youtube siva reddy) - FRIDAY DEADLINE

2) explore JIT -

<https://www.geeksforgeeks.org/jvm-works-jvm-architecture/>

3) explore the class file of any existing code and check it.

If we will directly open this file we cant understand even we have written it from our own .

We need a decompiler to read whats inside a .class file or we can use cmd

**C:\Users\coditas\Desktop>javap HelloWorld**

**Compiled from "HelloWorld.java"**

**class HelloWorld {**

**HelloWorld();**

**public static void main(java.lang.String[]);**

**}**

4) difference between access specifier and access modifier?

In older languages C /C++ (public ,private ,default,protected) are considered as Access specifiers and remaining are considered as Access modifiers

But in java there is no such type of division all are considered as Access modifiers.

(Public ,private ,default,protected,final ,strictfp,abstract,)

6) can we have multiple main methods in class?

Yes we can have multiple main method but only with overloaded main method.

7) Can we overload and override the main method?

Yes we can have overload main method but

**There should be only one main method with parameter as string[] arg.**

[Link](https://www.geeksforgeeks.org/can-we-have-multiple-main-methods-in-java/#:~:text=From%20the%20above%20program%2C%20we,parameter%20as%20string%5B%20%5D%20arg.)

We cannot override main method of java because a static method cannot be overridden.

Static methods are associated with class lets say we are calling static method with class name there will be ambiguity in finding the method in between base class and derived class method .

[Link](https://www.geeksforgeeks.org/how-to-overload-and-override-main-method-in-java/#:~:text=No%2C%20we%20cannot%20override%20main,is%20associated%20with%20an%20object.)

8) Can I write the main method as private, protected and default?

its fine to have private,protected or default at the time of compilation it’ll compile but runtime Error will occurs.

You can define the main method in your program without private, protected or, default (none) modifier, the program gets compiled without compilation errors.But, at the time of execution JVM does not consider this as the entry point of the program.

**C:\Users\coditas\Desktop>javac HelloWorld.java**

**Here compiles successfully..but at runtime Error**

**C:\Users\coditas\Desktop>java HelloWorld**

**Error: Main method not found in class HelloWorld, please define the main method as:**

**public static void main(String[] args)**

**or a JavaFX application class must extend javafx.application.Application**

9) without a main method can we execute our code? How?

Yes, we can only using static block because it will be handled by compiler

class PrintWithoutMain {

// static block

static

{

// prints "Hello World!!" to the console

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

// exit from the program

System.exit(1);

}

}

**The above code will work only in java 1.6 and older version.**

[link](https://www.geeksforgeeks.org/java-program-to-print-any-statement-without-using-the-main-method/?ref=rp)

10) Can we change the return type of the main() method from void to any other data type

like int return?

**NO**

**here it will get compile successfully**

**At runtime it will start showing errors like:**

**C:\Users\coditas\Desktop>javac HelloWorld.java**

**C:\Users\coditas\Desktop>java HelloWorld**

**Error: Main method must return a value of type void in class HelloWorld, please**

**define the main method as:**

**public static void main(String[] args)**

**C:\Users\coditas\Desktop>**

11) explore keyword strictfp?

Floating-point calculations are platform-dependent i.e. different output(floating-point values) is achieved when a class file is run on different platforms(16/32/64 bit processors). To solve this type of issue, strictfp keyword was introduced in JDK 1.2 version by following [IEEE 754](https://en.wikipedia.org/wiki/IEEE_floating_point) standards for floating-point calculations.

[link](https://www.geeksforgeeks.org/strictfp-keyword-java/)