

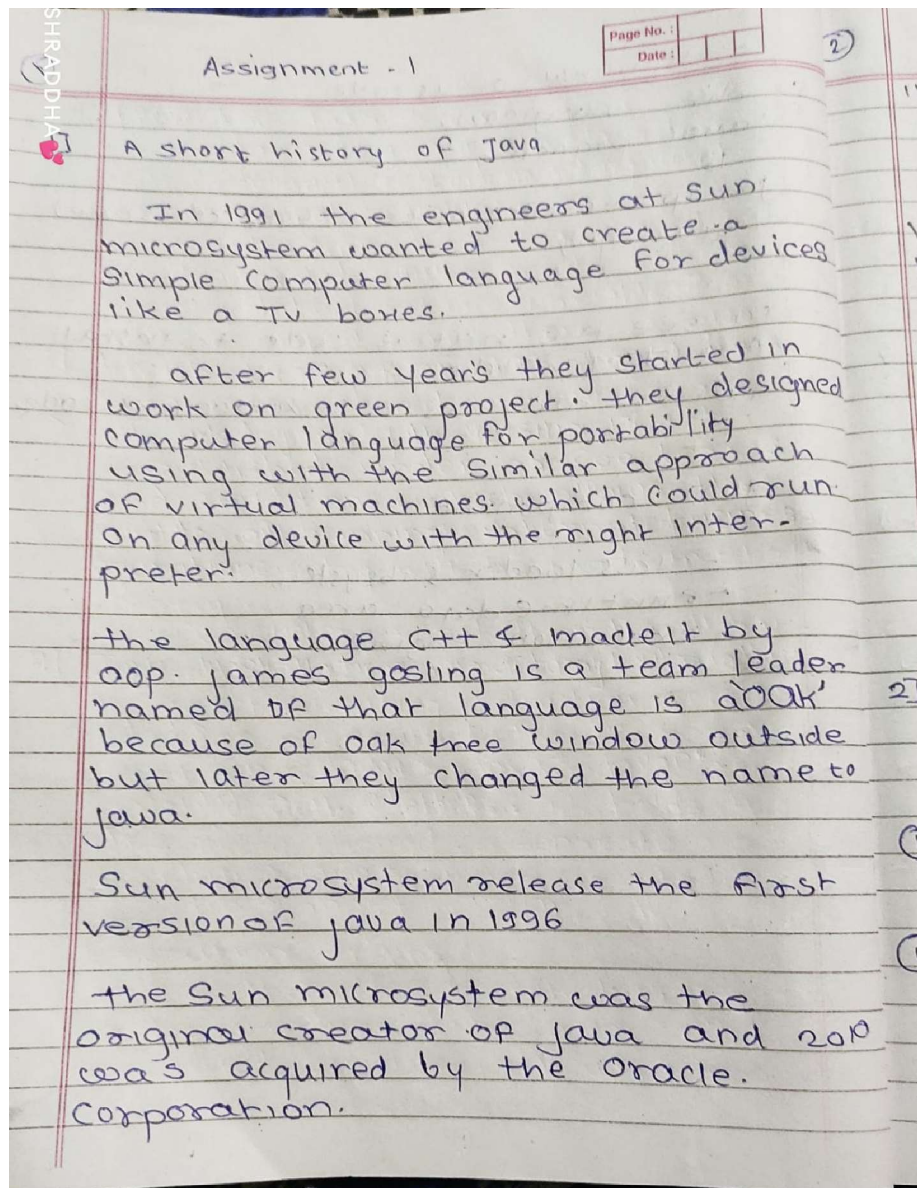
ASSGNMENT_1

1. Reading Assignment: A Short History of Java .

- Task: Read about the history and development of Java.

Link:

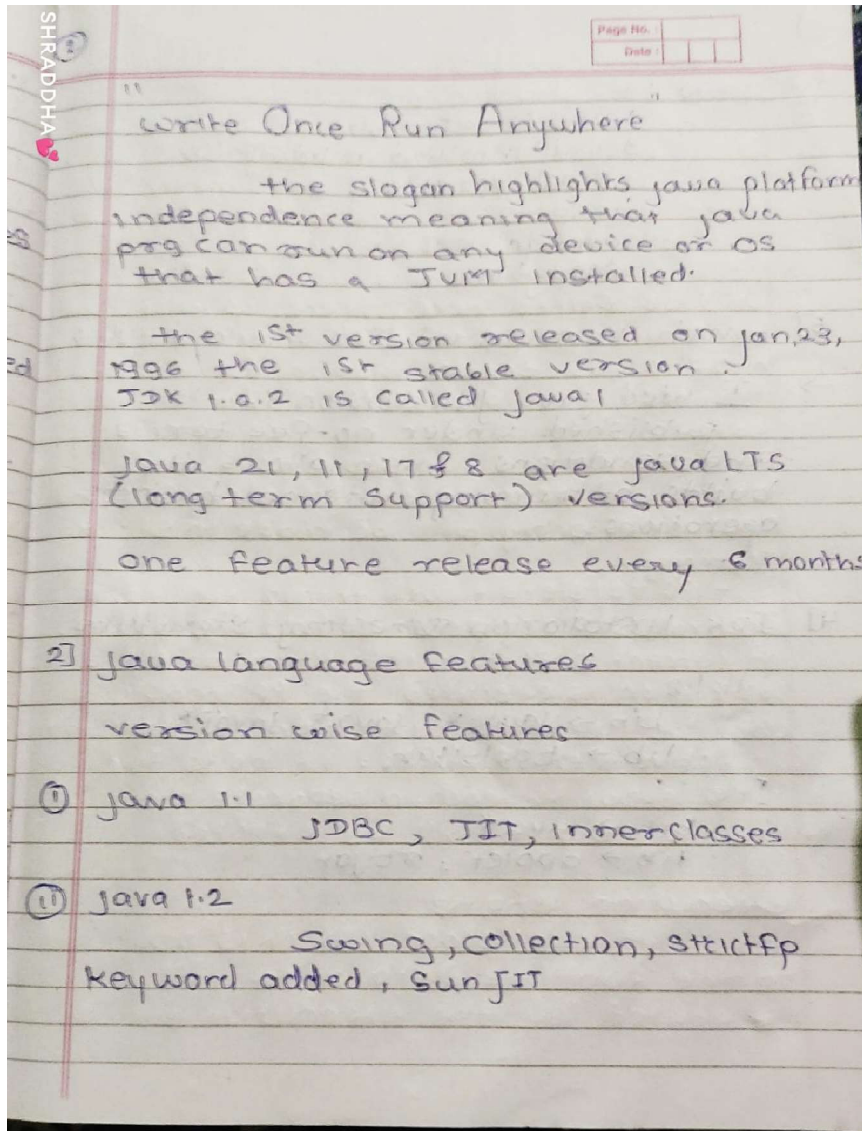
<http://sunsite.uakom.sk/sunworldonline/swol-07-1995/swol-07-java.html>



2. Reading Assignment: Java Language Features

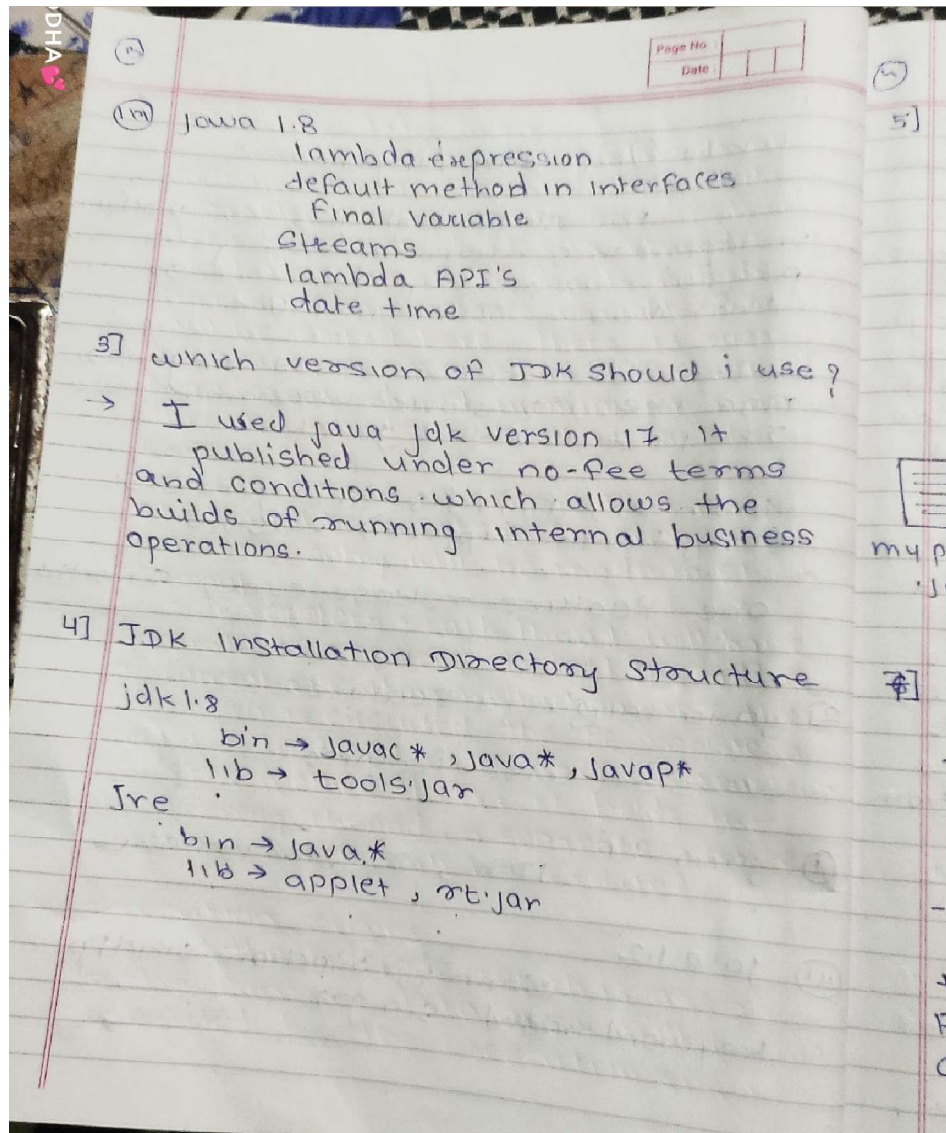
Task: Learn about the main features of Java.

Link: <https://javaalmanac.io/features/>



3. Reading Assignment: Which Version of JDK Should I Use? .

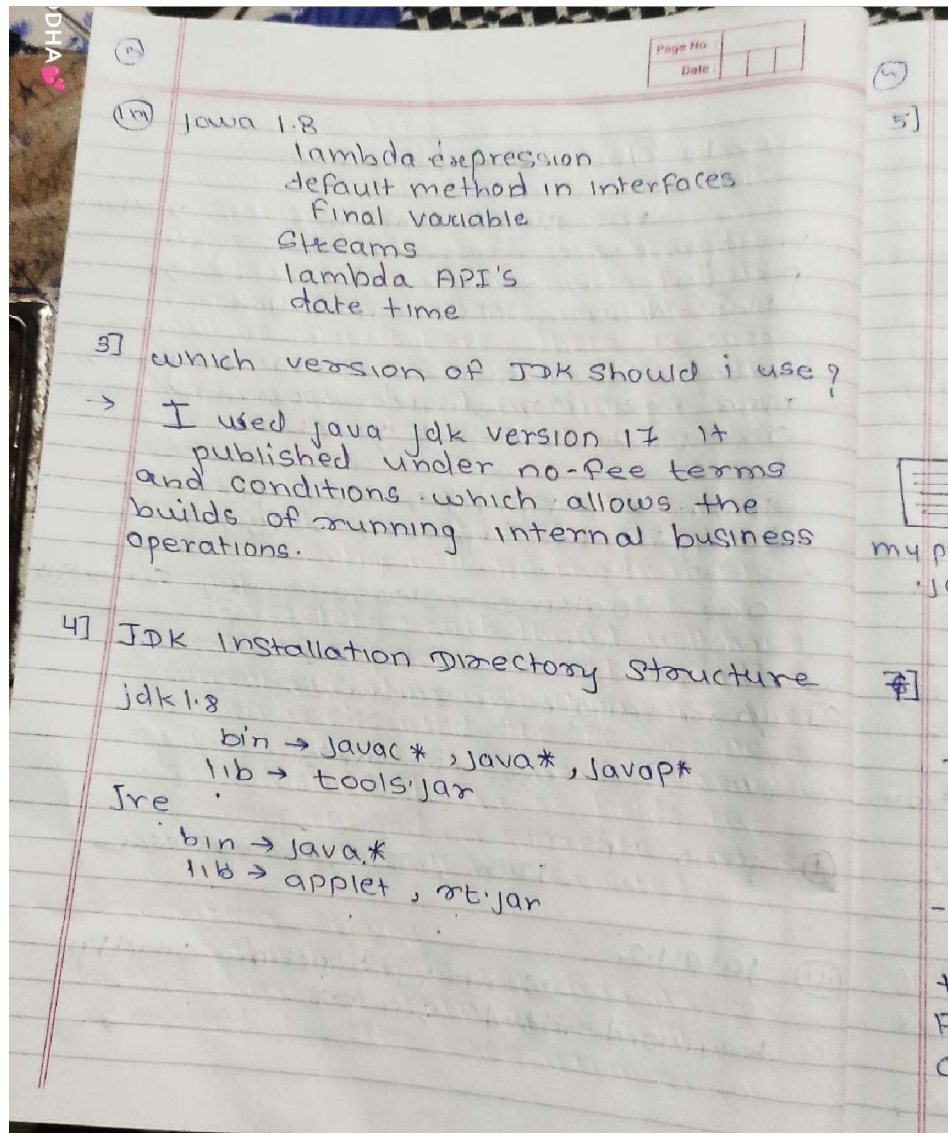
Task: Find out which JDK version is right for you. Link: <https://whichjdk.com/>



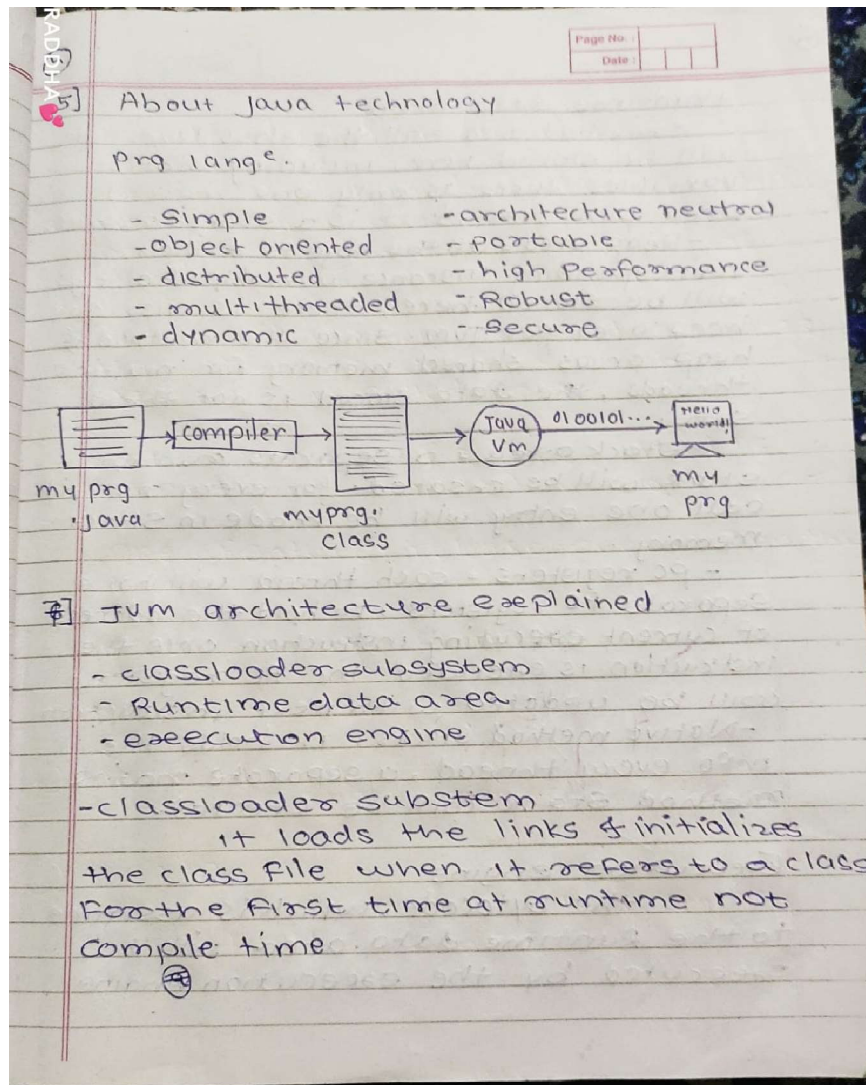
4. Reading Assignment: JDK Installation Directory Structure •

Task: Understand the folder structure and files in the JDK installation.

Link: <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jdkfiles.html>

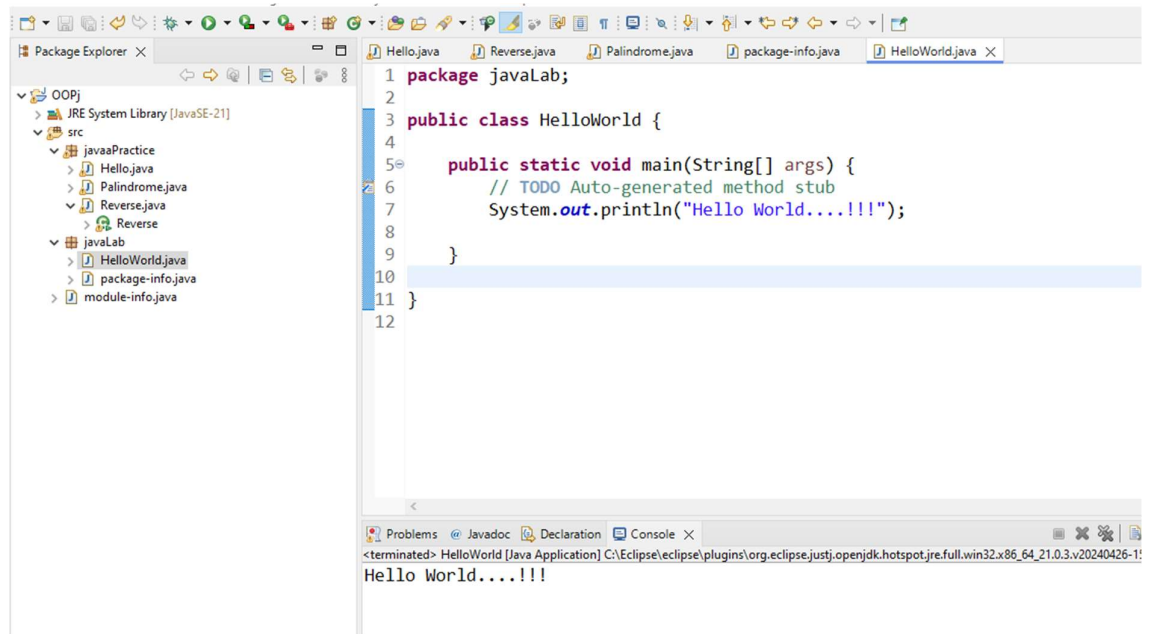


5. Reading Assignment: About Java Technology
Task: Read about the basics of Java technology and its components.
Link: <https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>

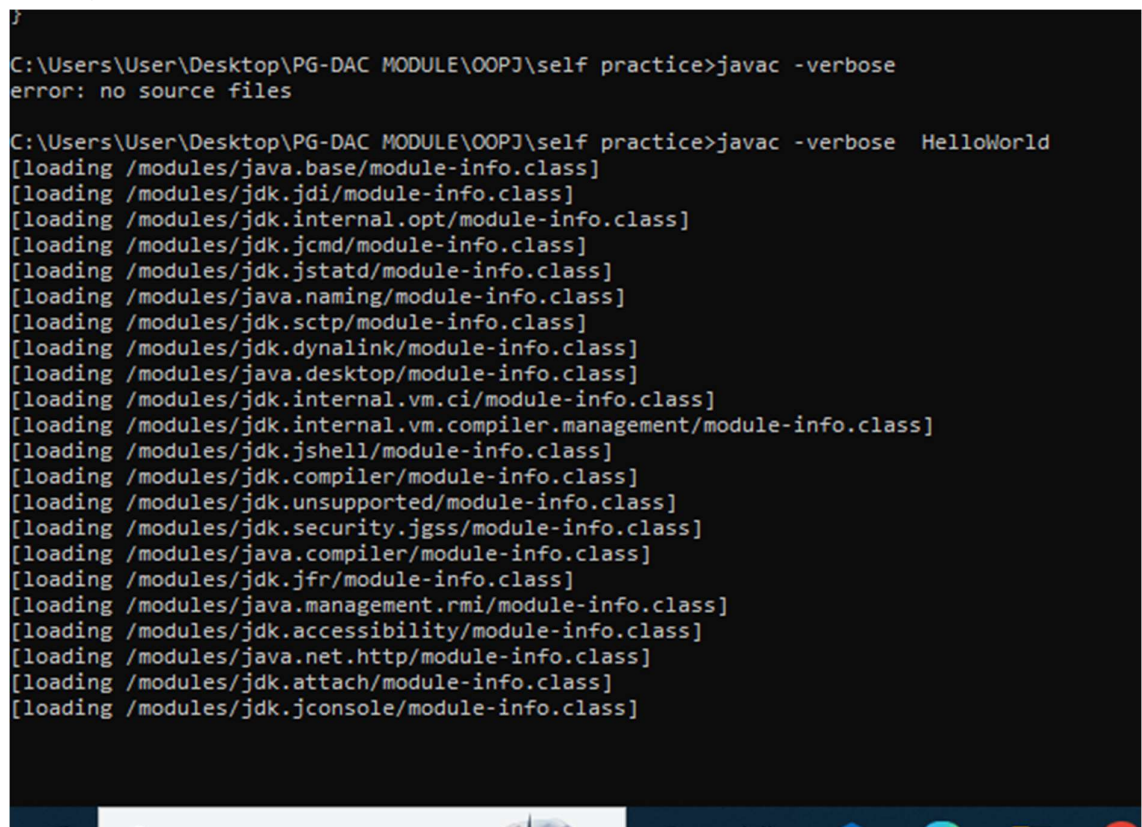


6. Coding Assignments

1. Hello World Program: Write a Java program that prints "Hello World!!" to the console.



2. Compile with Verbose Option: Compile your Java file using the with javac . Check the output.



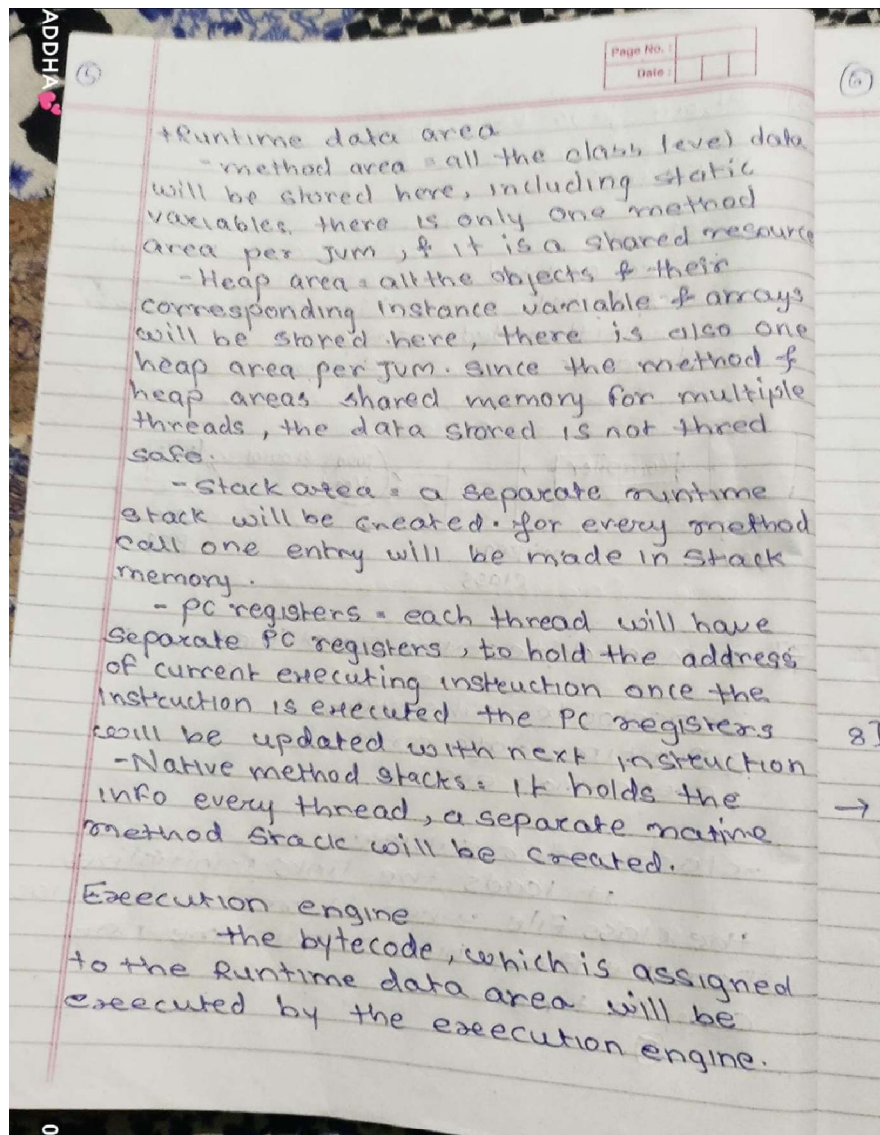
3. Inspect Bytecode: Use the `javap-verbose` option tool to examine the bytecode of the compiled `.class` file. Observe the output.

```
C:\Users\User\Desktop\PG-DAC MODULE\OOPJ\self practice>javap HelloWorld
Compiled from "HelloWorld.java"
public class HelloWorld {
    public HelloWorld();
    public static void main(java.lang.String[]);
}
C:\Users\User\Desktop\PG-DAC MODULE\OOPJ\self practice>
```

7. Reading Assignment: The JVM Architecture Explained

Task: Learn about how the Java Virtual Machine (JVM) works.

Link: <https://dzone.com/articles/jvm-architecture-explained>



8. Reading Assignment: The Java Language Environment: Contents .

Task: Explore the content and features of the Java language environment. Link:

<https://www.oracle.com/java/technologies/language-environment.html>

It reads the bytecode & executes it piece by piece.

Interpreter - it interprets the bytecode faster but executes slowly. the disadvantage of the interpreter is that when one method is called multiple times, every time a new interpretation is required.

JIT compiler - it neutralizes the disadvantage of the interpreter. the execution engine will be using the help of the interpreter in converting the byte code but when it finds repeated code it uses the JIT compiler, which compiles the entire bytecode & changes it to native code, the native code will be used directly for repeated method calls which improve the performance of the system.

8] The java language environment: Contents.

- architectural & neutral - java technology is designed to support application that will be developed into heterogeneous n/w env. applⁿ must be capable of executing on a variety of h/w architectures. with this variety of hardware platforms applⁿ must execute atop of a variety of OS & interoperate with multiple programming language interface.