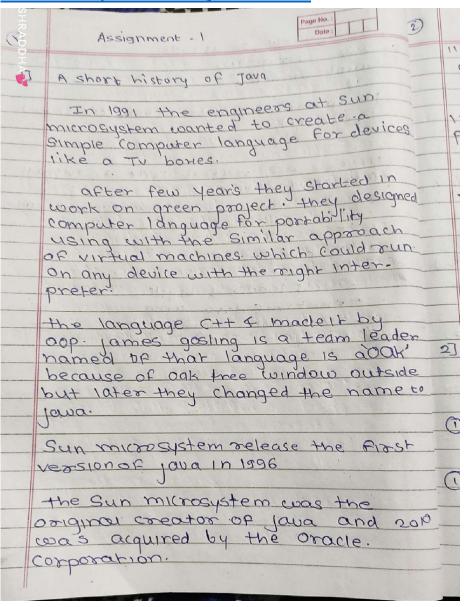
## ASSGNMENT\_1

- 1. Reading Assignment: A Short History of Java
  - Task: Read about the history and development of Java.

Link:

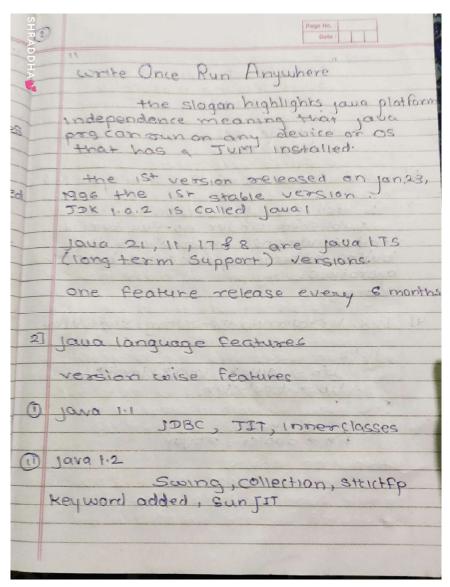
http://sunsite.uakom.sk/sunworldonline/swol-07-1995/swol-07-java.html



2. Reading Assignment: Java Language Features

Task: Learn about the main features of Java.

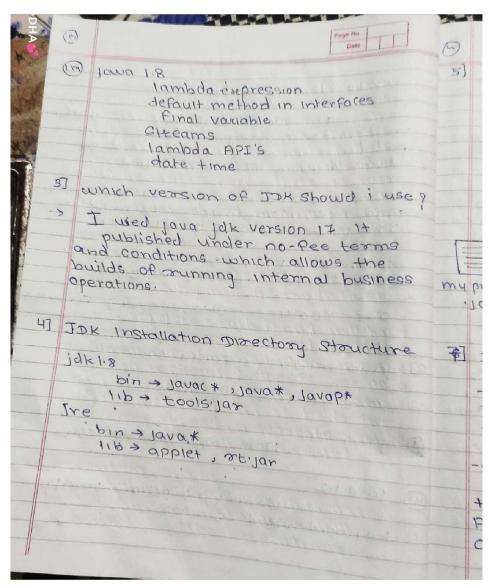
Link: <a href="https://javaalmanac.io/features/">https://javaalmanac.io/features/</a>



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

Task: Find out which JDK version is right for

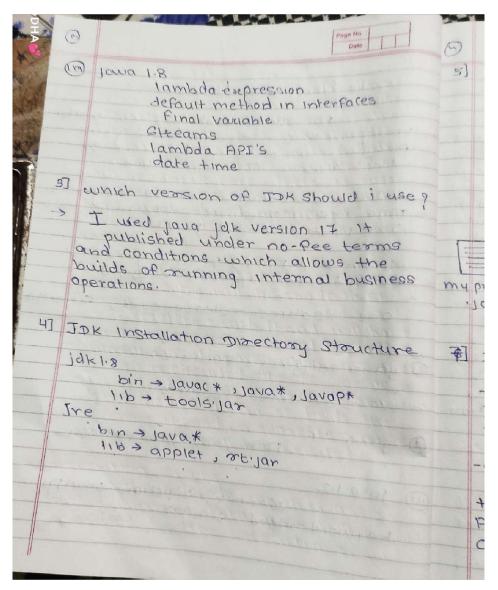
you. Link: https://whichjdk.com/



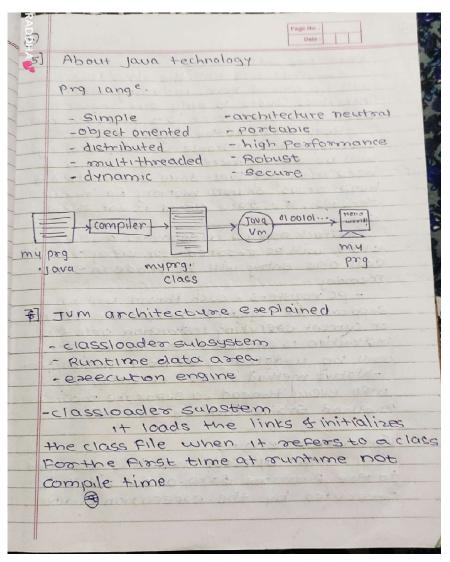
4. Reading Assignment: JDK Installation Directory Structure ●

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

Link:https://docs.oracle.com/javase/8/docs/t echnotes/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.

```
Package Explorer X
                                □ □ I Hello,java □ Reverse.java □ Palindrome.java □ package-info,java □ HelloWorld.java ×

⇔ ⇔ @ | E ⊗ | № § | 1 package javaLab;

√ № 00Pj

 > M JRE System Library [JavaSE-21]
                                        3 public class HelloWorld {
   ✓ ∰ javaaPractice
                                               public static void main(String[] args) {
     > 🕖 Hello.java
      Palindrome.java
                                                   // TODO Auto-generated method stub
     ✓ ☑ Reverse.java
                                                   System.out.println("Hello World....!!!");
       > 🔐 Reverse

→ iavaLab

     >  HelloWorld.java
   10
                                      11 }

    Problems @ Javadoc    Declaration    □ Console ×
                                                                                                                  m × % |
                                      cterminated> HelloWorld [Java Application] C:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1
                                      Hello World....!!!
```

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

```
C:\Users\User\Desktop\PG-DAC MODULE\OOPJ\self practice>javac -verbose
error: no source files
C:\Users\User\Desktop\PG-DAC MODULE\OOPJ\self practice>javac -verbose HelloWorld
[loading /modules/java.base/module-info.class]
[loading /modules/jdk.jdi/module-info.class]
[loading /modules/jdk.internal.opt/module-info.class]
[loading /modules/jdk.jcmd/module-info.class]
[loading /modules/jdk.jstatd/module-info.class]
[loading /modules/java.naming/module-info.class]
[loading /modules/jdk.sctp/module-info.class]
[loading /modules/jdk.dynalink/module-info.class]
[loading /modules/java.desktop/module-info.class]
[loading /modules/jdk.internal.vm.ci/module-info.class]
[loading /modules/jdk.internal.vm.compiler.management/module-info.class]
[loading /modules/jdk.jshell/module-info.class]
[loading /modules/jdk.compiler/module-info.class]
[loading /modules/jdk.unsupported/module-info.class]
[loading /modules/jdk.security.jgss/module-info.class]
[loading /modules/java.compiler/module-info.class]
[loading /modules/jdk.jfr/module-info.class]
[loading /modules/java.management.rmi/module-info.class]
[loading /modules/jdk.accessibility/module-info.class]
[loading /modules/java.net.http/module-info.class]
[loading /modules/jdk.attach/module-info.class]
[loading /modules/jdk.jconsole/module-info.class]
```

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

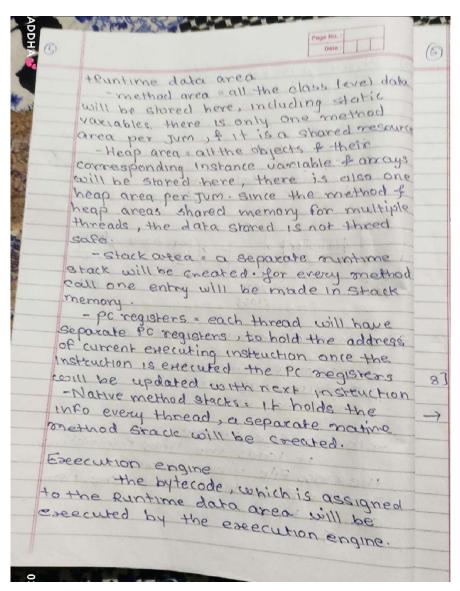
```
C:\Users\User\Desktop\PG-DAC MODULE\OOPJ\self practice>javap HelloWorld
Compiled from "HElloWorld.java"
public class HelloWorld {
   public HelloWorld();
   public static void main(java.lang.String[]);
}
C:\Users\User\Desktop\PG-DAC MODULE\OOPJ\self practice>
```

7. Reading Assignment: The JVM Architecture Explained

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

Link: <a href="https://dzone.com/articles/jvm-">https://dzone.com/articles/jvm-</a>

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 12

interpreter - it interprets the bytecode piece by piece. Paster but executes Slowly. the disadv of the interpreter is that when one method is called multiple times, every time a new interpretation is required TIT compiler - 1+ neutralizes the disadv . 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 JJT compiler, which compiles the entire byte code & 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.

  To acchitectural & newtral jawa technology

  15 designed to support application

  15 designed to support application

  15 that will be developed into heterogenous

  16 that will be developed into heterogenous

  17 that will be developed into heterogenous

  18 that will be developed into heterogenous

  18 that will be capable of

  18 executing on a variety of him architects withis this variety of hardware

  18 plaforms apply must execute atop of a

  18 plaforms apply must execute atop of a

  18 variety of Os Interoperate with

  18 multiple programing language interface,