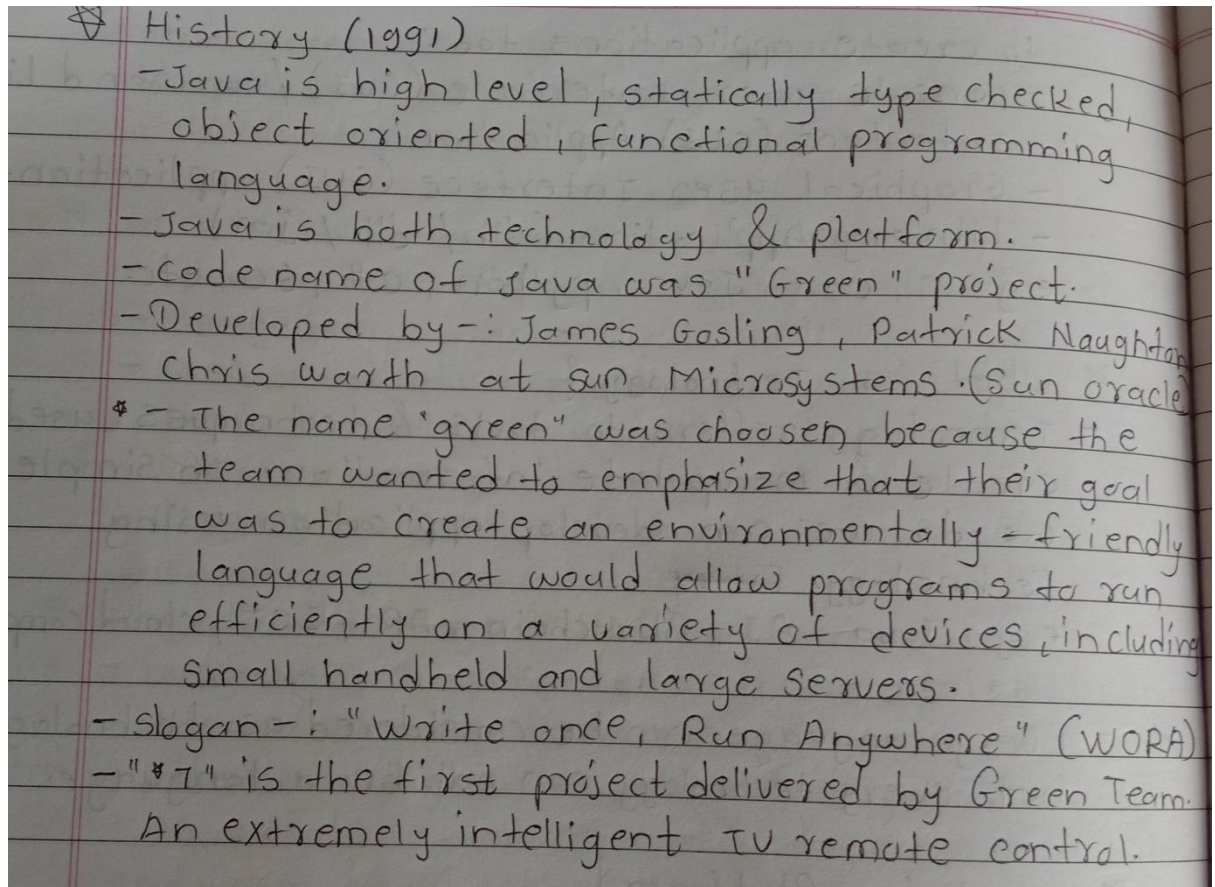


Assignment 1

1. Reading Assignment: A Short History of Java



2. Reading Assignment: Java Language Features

1) Features of Java Programming language.

- most popular & commonly used programming lang.
- widely known as for performance, platform independence and security.

↳ It is inspired by C/C++. most of the syntax is similar to these language.

2) object-oriented

- Java is object oriented programming lang.
- It supports every oop concept such as.

Sandeepkulange@gmail.com

Abstraction, Inheritance, Encapsulation & polymorphism. `main()` function is define under a class in Java.

3) Platform independence

- Java is platform independence because it supports WORA means Write Once Read Anywhere. Program can develop application in one operating system & run on any other without modification.

4) Compile and interpreted

- when java program is created, the Java compiler (javac) compiles the java source code into byte code. The Java virtual Machine (JVM) is an interpreter that convert byte code to machine code. which is portable & can be executed on any operating system.

5) Dynamic

- It is more dynamic to C & C++. It allow programmer to link new class libraries, objects & methods dynamically. It also have a large run time information.

6) Security

- It is a secure programming language. Java also ensure that no viruses enter an applet.

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

4. Reading Assignment: JDK Installation Directory Structure

jdk/Contents/Home . The root directory of the JDK software installation. This directory also contains copyright, README, and src. zip files, which is the source code archive file of the Java platform.

5. Reading Assignment: About Java Technology

The Java programming language is a high-level language

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to processor it instead contains bytecodes the machine language of the Java Virtual Machine (Java VM). The java launcher tool then runs application with an instance of the Java Virtual Machine.

A *platform* is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

The Java Virtual Machine

The Java Application Programming Interface (API)

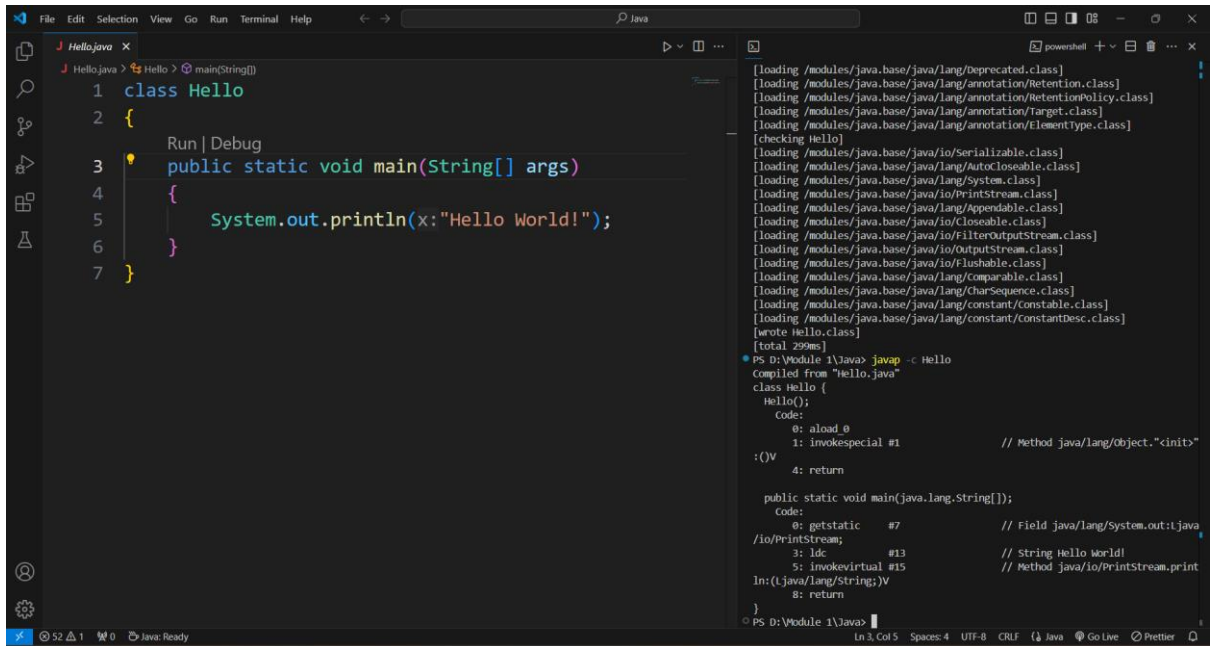
Java Virtual Machine is the base for the Java platform and is ported onto various hardware-based platforms.

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as *packages*.

6. Coding Assignments

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

ASSIGNMENT NO.1



The screenshot shows an IDE with two panels. The left panel displays a Java file named 'Hello.java' with the following code:

```
1 class Hello
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(x:"Hello World!");
6     }
7 }
```

The right panel shows the output of the program, which is a list of loaded classes and the execution of the main method:

```
[loading /modules/java.base/java/lang/Deprecated.class]
[loading /modules/java.base/java/lang/annotation/Retention.class]
[loading /modules/java.base/java/lang/annotation/RetentionPolicy.class]
[loading /modules/java.base/java/lang/annotation/Target.class]
[loading /modules/java.base/java/lang/annotation/ElementType.class]
[checking Hello]
[loading /modules/java.base/java/io/Serializable.class]
[loading /modules/java.base/java/lang/AutoCloseable.class]
[loading /modules/java.base/java/lang/System.class]
[loading /modules/java.base/java/io/PrintStream.class]
[loading /modules/java.base/java/lang/Appendable.class]
[loading /modules/java.base/java/io/Closeable.class]
[loading /modules/java.base/java/io/FilterOutputStream.class]
[loading /modules/java.base/java/io/OutputStream.class]
[loading /modules/java.base/java/io/Flushable.class]
[loading /modules/java.base/java/lang/Comparable.class]
[loading /modules/java.base/java/lang/CharSequence.class]
[loading /modules/java.base/java/lang/constant/Constable.class]
[loading /modules/java.base/java/lang/constant/ConstantDesc.class]
[wrote Hello.class]
[total 299ms]
PS D:\Module 1\Java> javap -c Hello
Compiled from "Hello.java"
class Hello {
    Hello();
    code:
        0: aload 0
        1: invokespecial #1          // Method java/lang/Object.<init>()V
        4: return
    public static void main(java.lang.String[]);
    code:
        0: getstatic #7              // Field java/lang/System.out:Ljava
        /io/PrintStream;
        3: ldc #13                   // String Hello World!
        5: invokevirtual #15         // Method java/io/PrintStream.print
        ln:(Ljava/lang/String;)V
        8: return
}
```

7. Reading Assignment: The JVM Architecture Explained

The JVM is responsible for interpreting the bytecode and translating it into machine code that can be executed by the underlying operating system. It also provides memory management, security, and other runtime services necessary for executing Java applications.