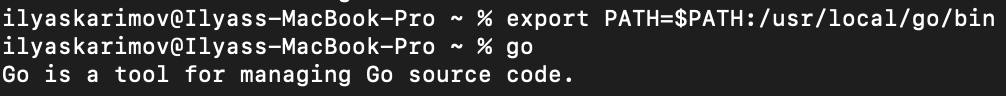
**Advanced Software Paradigms**

Assignment 1. Ilyas Karimov

Task 4.

**Go:**

To install Go, I used this instruction/Link: [https://golang.org/doc/install?download=go1.15.2.darwin-amd64.pkgA screenshot of a cell phone

Description automatically generated](https://golang.org/doc/install?download=go1.15.2.darwin-amd64.pkg)

**Scala:**

To install Scala, I used this instruction:

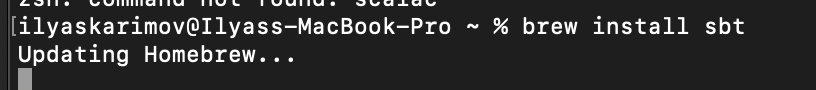
<https://www.scala-lang.org/download/>

To use Scala, I made sure I have minimum version 1.8 or 11 installed. I downloaded

A screenshot of a cell phone

Description automatically generated

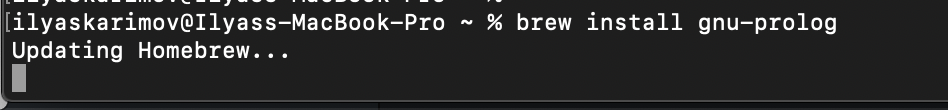
I installed scala compiler by running **brew install sbt.**



**SWI-Prolog:**

To install Prolog, I used this instruction:

<https://www.swi-prolog.org/download/stable>

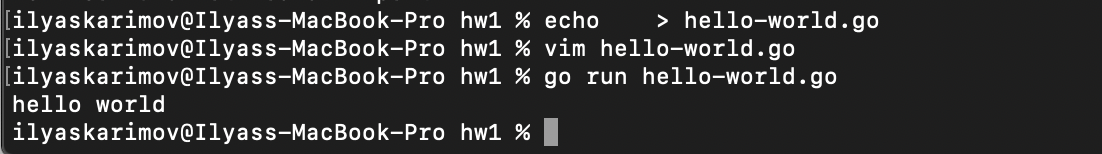
I installed prolog by running **brew install gnu-prolog**. 

**Compiling/Source Codes:**

**Go:**

I used the Command-Line tool to compile the code, but I read that I could have used Microsoft Visual Code as an IDE for Go. I installed version go1.15.2 darwin/amd64. File extension is **.go**

Reference: <https://gobyexample.com/hello-world>



**Source Code in hello-world.go file**:

import "fmt"

func main() {

fmt.Println("hello world")

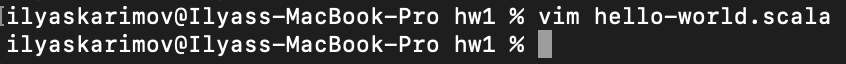
}

Explanation:

1. Fmt comes from the word format, since Go was inspired a lot from C. fmt performs I/O functions, specifically, scanf() and printf() in C.
2. Func stands for function and it is accepted as a type in Go like in JavaScript.
3. But what I realized is semi-colons (;) are not used in Go language to end the statements.

**Scala:**

I downloaded IntelliJ for Scala, but then I had some problems, consequently, I used command-line. Installed version is 2.13.3. File extension **.scala**



**Source Code in hello-world.scala file**:

object ScalaHelloWorld {

def main(args: Array[String]): Unit = {

println("Hello World")

}

}

Explanation:

1. I learnt that **object** is a named instance with members such as fields and methods. Object and class have the same name.
2. It looks like Java. Def here means a function; we pass the argument. As far as I understood, Unit is a type like Void, and here, no value is to be returned.
3. Another difference is no semi-colons are used for end statements.
4. Rules for curly braces are the same {}.

A screenshot of a cell phone

Description automatically generated

**Prolog:**

Reference: <https://www.youtube.com/watch?v=SykxWpFwMGs>

I used Command-Line for prolog as well. Normally, we store variables and operate on them, but in Prolog, I learnt that we store relationships in prolog. File extension is **.pl**

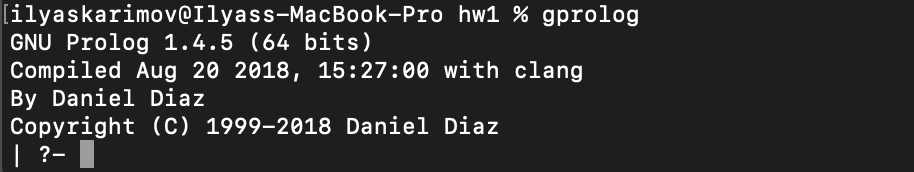
**Source Code in hello-world.pl file**:

write('Hello World'), nl.

Explanation:

1. Write means printing and nl means new line.
2. The statement ends with the period sign (.)

I also learnt that in order to stop the process, I need to use the command **halt.**



A screenshot of a cell phone

Description automatically generated

P.S. I wrote the code in hello-world.pl file and it showed that I can run the file with the consult command, yet it gave me errors. The same code worked like a charm in gprolog. So, I submitted that, the right one is above, the problem that I had is below:

