{Difference b/w C & Java}

```
#include <stdio.h>
int main()
{
    printf("ECE_EEE");
    return 0;
}
```

Demo.c

main function

DateOfBirth (Demo.java)

Pascal case ↑ RateOfInt

```
public class Demo {
    public static void main
        (String[] args){
        System.out.println("ECE");
    }
}
```

camel Case

main Method

rateOfInt

methods, vars

① JDK → Version (17.0) (Platform Dependent) (W/M/L/U)

② (env vars) → path : C/Program Files/Java/JDK-17.0

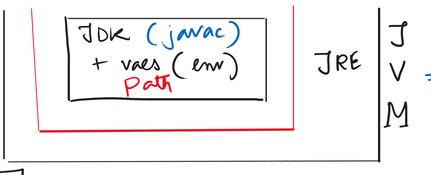
JAVA_HOME

Runtime Environment → JRE

JDK + Path = JRE = Java Runtime Environment

↓

javac



file → Demo.java

javac Demo.java

compilation

Demo

Symbolic Language ← Demo.class [Bytecode]

ABC.java.txt

View → slow

[Demo.java]

RUN Code:

java Demo

JDK+JRE - JVM (Java Virtual Machine)

(WORA) ⇒ Write Once Run Anywhere!

Why is Java platform independent?

[.java ⇒ .class] [Sym Lang]

JDK+JRE

JVM

(javac)

Any op Sy { W-M, M-L, L-W }

JIT → JVM

Just In Time Compiler: java code? native

String[] args is an array of strings used to pass Command Line Arguments to the main method to perform :→

(Sanity Testing)

* If our main is not working, the app will not run. So we test it.