Dt : 19/11/2020

***imp**

define 'Program'?

=>Program is a 'Set-of-Instructions'.

Note:

=>After writing the program,we save the program with Language extention.

Exp:

Test.c

Test.cpp

Test.java

=>The program will have the following two stages:

1.Compilation Stage

2.Execution Stage

1.Compilation Stage:

=>The process of checking the program constructed within the rules of language or not,is known as Compilation process.

Note:

=>If the compilation process is successfull then,

=>c and c++ languages generate Objective Code

=>Java Language generate Byte Code

=>c,c++ and Java lanaguages will use Compiler in compilation process.

2.Execution Stage:

=>The process of running the compiled code and checking the required output is generated or not,is known as Execution process.

=>This execution process internally having the following two SubProcesses:

(i)Loading process

(ii)Linking process

(i)Loading process:

=>The process of loading the required files into current running program using 'Loader' is known as Loading process.

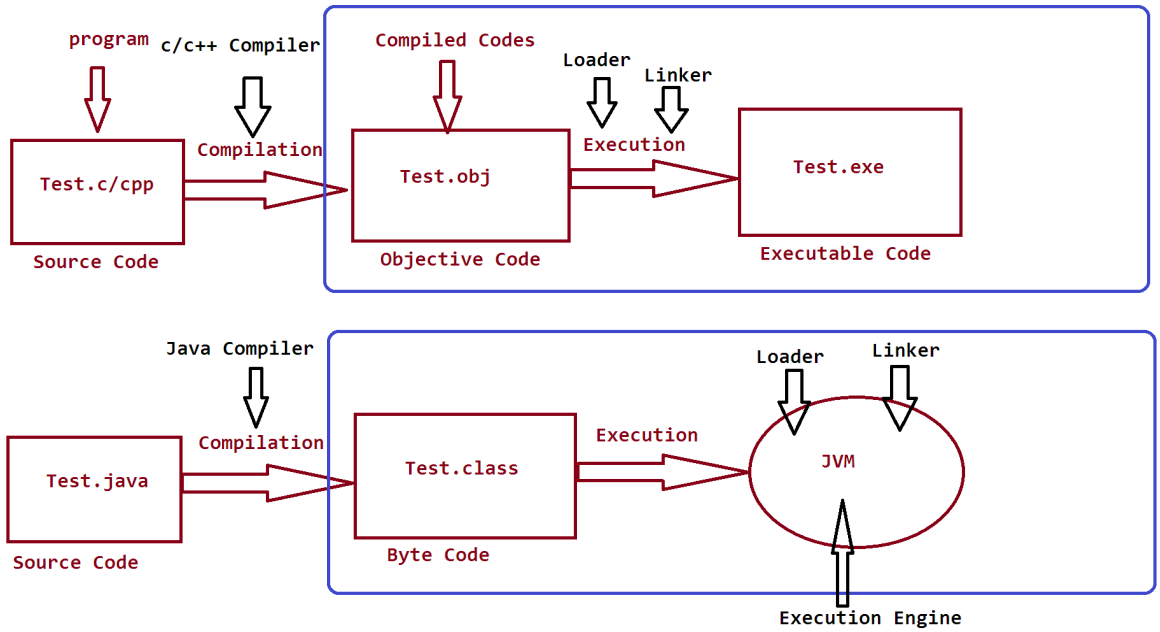
(ii)Linking process:

=>The process of linking the loaded files into current running program using 'Linker' is known as Linking process.

Note:

=>In c and c++,after loading and linking process is successfull then the Objective code is converted into executable code and which is executed.

=>In Java language the byte code is executed on JVM(Java Virtual Machine) and which internally having Loader,Linker and ExecutionEngine.



Summary:

step1 : write the program in Java and save with language extention as follows

Class_name.java

Exp:

Test.java

step2 : Compile the program using Java Compiler and after compilation process 'Byte code' is generated.

Exp:

Test.class

step3 : Execute the Byte code on JVM(Java Virtual Machine)

==========

Dt : 20/11/2020

faq:

wt is the diff b/w

(i)Objective Code

(ii)Byte Code

(i)Objective Code:

=>Objective code is a compiled code generated from c and c++ programs.

=>while Objective Code generation the Operating System(OS) is involved,because of this reason Objective Code is PlatForm dependent Code.

DisAdvantage:

=>The Objective Code which is generated from one PlatForm cannot be executed on other PlatForms.

Note:

=>c and c++ languages which are generating objective code are PlatForm dependent languages and not preferable for Internet application development.

(ii)Byte Code:

=>Byte code is a compiled code generated from Java programs.

=>while Byte Code generation the Operating System(OS) is not involved,because of this reason Byte Code is PlatForm independent code.

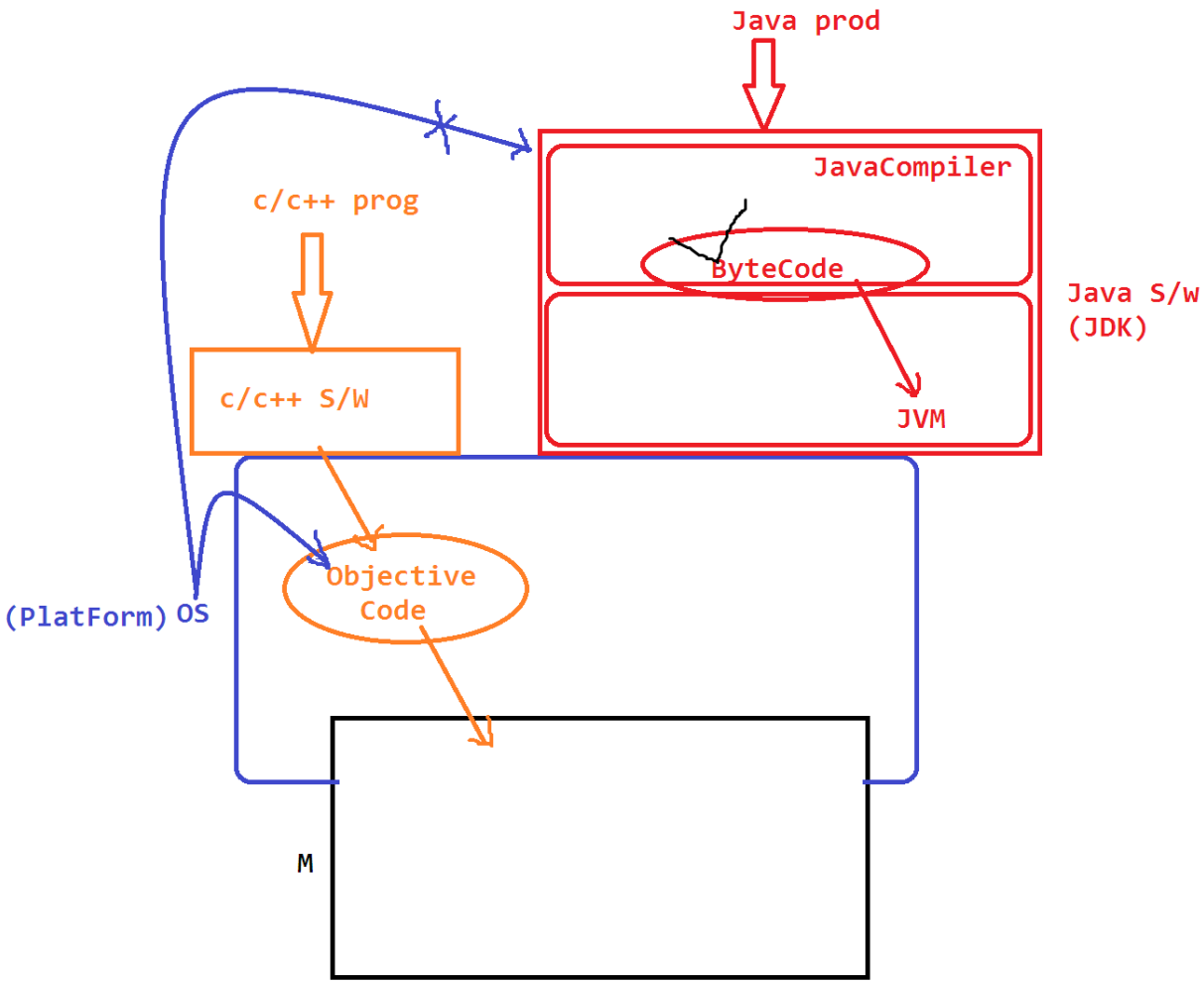
Advantage:

=>The Byte code which is generated from one PlatForm can be executed

on all the PlatForms.

Note:

=>The Java Language which is generating Byte code is PlatForm Independent language and which is preferable for Internet application development.



==========

Dt : 23/11/2020

Java History:

=> James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of sun engineers called Green Team.

=> Initially designed for small, embedded systems in electronic appliances like set-top boxes.

=> Firstly, it was called "Greentalk" by James Gosling, and the file extension was .gt.

Exp:

ProgName.gt

=>The language named as "Java" and released first version by 1995.

Java Versions:

1995 - Java Alpha&Beta

1996 - JDK 1.0

1997 - JDK 1.1

1998 - JDK 1.2

2000 - JDK 1.3

2002 - JDK 1.4

2004 - Java5(Tiger Java)

=>JDK 1.5

=>JRE 1.5

2006 - Java6

=>JDK 1.6

=>JRE 1.6

2011 - Java7

=>JDK 1.7

=>JRE 1.7

2014 - Java8

=>JDK 1.8

=>JRE 1.8

2017 - Java9

=>JDK 1.9

=>JRE 1.9

2018,2019,2020

->Java10,Java11,Java12,Java13,Java14

----

faq:

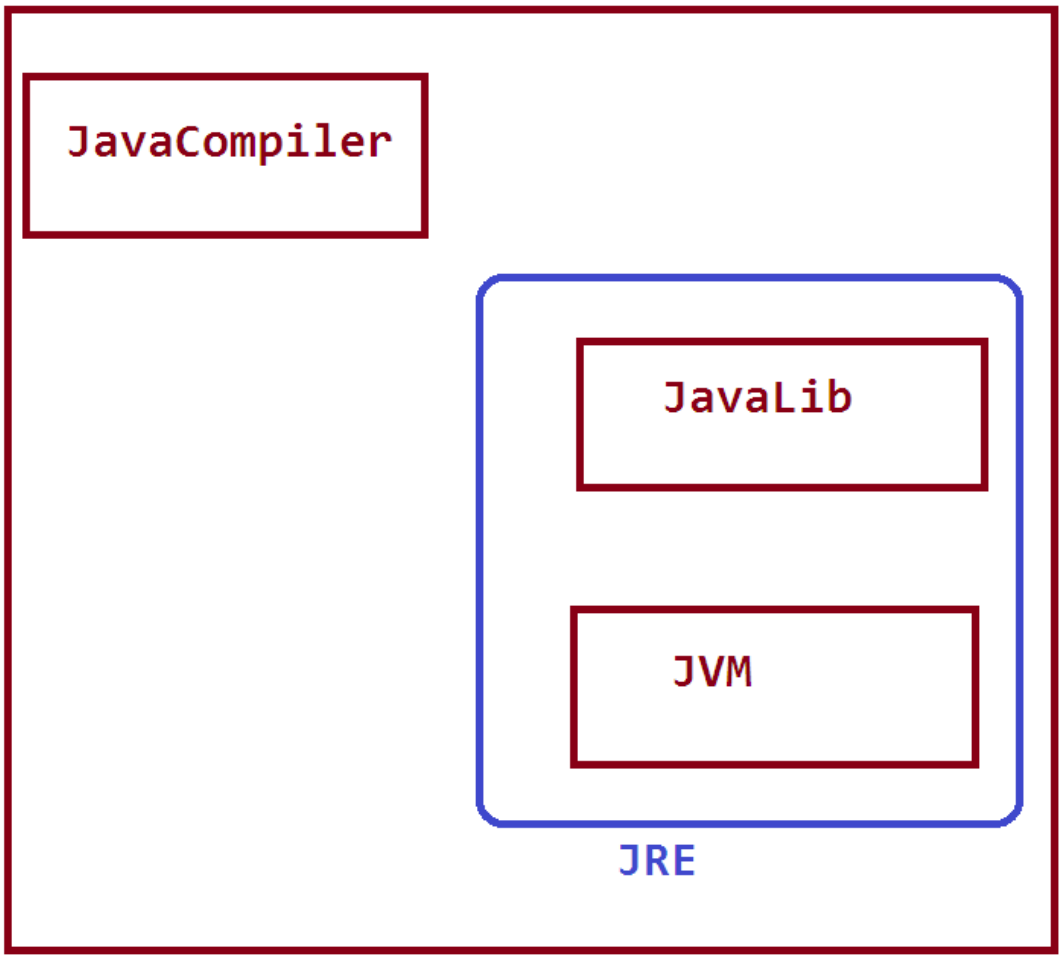
wt is the diff b/w

(i)JDK

(ii)JRE

(i)JDK:

=>JDK stands for 'Java Developer Kit' and which is collection of JavaCompiler,JavaLib and JVM.



JavaCompiler - which is used to compile the program and generate Byte code.

JavaLib - which provides built-in components used for program Construction.

JVM - which is used to execute Java Byte Code.

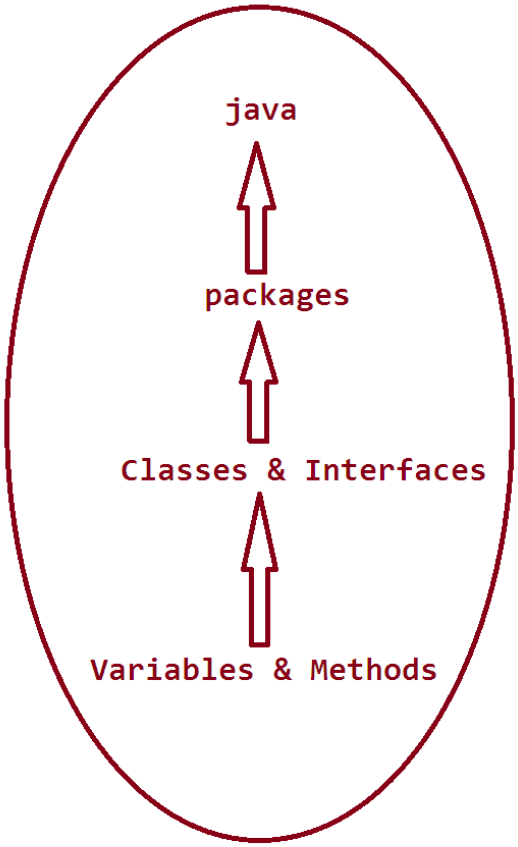
JavaLib:

=>JavaLib is represented with a word "java".

=>JavaLib is collection of "packages".

=>'packages' are collection of "classes and Interfaces".

=>'Classes and Interfaces' are collection of "Variables and Methods".



The following are some important packages from JavaLib:

java.lang - language package.

java.io - IO Stream and Files package

java.util - Utility package

java.net - Networking package

---

java.awt - Abstract Window Toolkit package(GUI)

javax.swing - Swing Programming package(GUI)

java.applet - Applet programming(GUI)

---

java.sql - DataBase Connection package

javax.servlet - Servlet programming package

javax.servlet.jsp - JSP programming package.

---

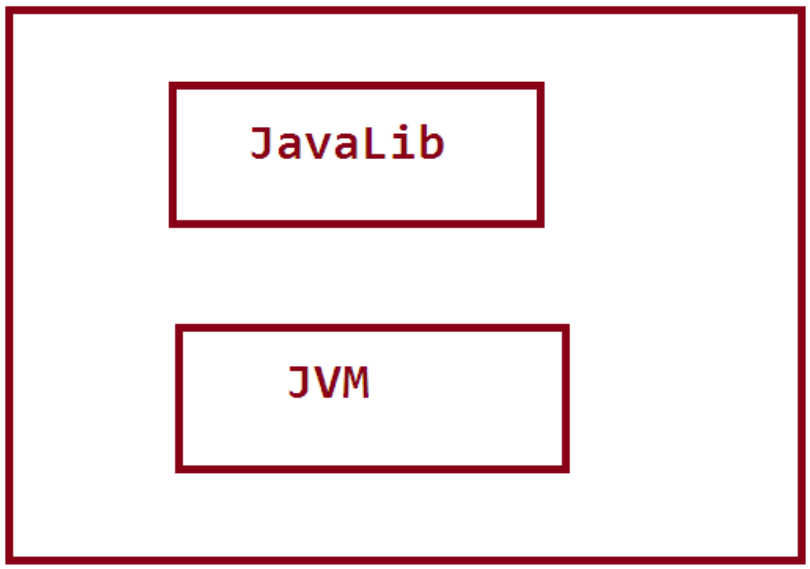
ii)JRE:

=>JRE Stands for 'Java Runtime Environment' and is collection of only JavaLib and JVM.

(JRE will not contain JavaCompiler)

Note:

=>In realtime JRE is used part of WebServers while executing WebAppl.



JRE

(Java Runtime Environment)

==========