JDK installation and setting environment

**How to install**

Before you install java (JDK) on your machine first check whether java is installed or not.

**To check java is installed or not-**

**1st Way-**

Go to “Start” 🡪 “Open Run” 🡪 “Enter ‘cmd’ and click on OK” 🡪 “Enter **‘java –version’** in command prompt and press enter

If java is installed then it will shows which java version is installed.

If it is not installed it will give an error like “***‘java’ is not recognizing as internal and external command***”.

**2nd Way-**

[**http://www.java.com/en/downloads/testjava.jsp**](http://www.java.com/en/downloads/testjava.jsp)

**How to download JDK.exe file-**

**How to install JDK-**

1. Double click on the JDK.exe
2. Click “Next” for installing jdk
3. Click continue button to finish installation.

**Why do we need to set the path of jdk**

We need to set the path to tell the OS where our java commands are present and recognize it.

**How to check path is set or not**

Command: **“javac” and press enter**

**How to set path-**

**There are two ways**

**1st way (Temporary) –**

Use following command in command prompt

**set path= “(paste path of the bin folder of jdk file)”.**

**2nd way (Permanent) –**

1. Open java bin folder in the installed path
2. Copy the path
3. Then go to my computer properties
4. Then click on Advanced system settings
5. Click on environment variables
6. Click on path in system variable
7. Click on edit and paste it there (Note : Before paste it go to the last of inbuilt path and add semicolon then paste it and end with semicolon).

**What is JDK, JRE and JVM**

**JDK**

JDK is an acronym Java Development Kit. It physically exists. It contains JRE and Development tools.

**JRE**

JRE is a run time environment where JVM executes the byte code and contains class libraries and supporting files.

**JVM**

It is an abstract machine where java byte code is executed. JVM’s are available for many hardware and software platforms (so JVM is platform dependant)

**JVM performs three actions**

1. **Load class**
2. **Verifies code**
3. **Execute code**

**Class loader-** Is a part of sub-system of JVM is used to load class files.

**Byte-code verifier-** Verifies the byte code if it contains anything illegal that can violate access right to objects.

**Interpreter-** It will read byte-code stream and then execute the instructions.

**1.5 How to compile .java file**

Command: **“javac source\_file\_name(.java file name) with extension(.java)”.**

**How to execute .class file**

Command: **“java .class\_file\_name (Without extension)”.**

**Development tools**

**e.g.**

**Java.exe**

**Javac.exe**

**javaDoc.exe**

**Set of (.jar) Libraries**

**JVM**

**Other files**

**JRE**

**JDK**

**Rules for saving .java file, for learners**

* Starting character of a class name should be in uppercase (Industry Convention). Note: It works even if you don’t use uppercase.
* Class name and the file name might be same.

**Minimum number of lines to compile java file**

Class Demo{

}

**Minimum number of lines to compile and execute java file**

Class Demo{

Public static void main(String args[]){

}

}

WAP which give output “I Love Java”

class Demo{

public static void main(String args[]){

System.out.println(“I Love Java”);

}

}

**Rules for naming a class**

* Class name must begin with a letter, underscore ( \_ ) or a dollar sign ($).
* Class name may contain only letters, digits, underscores and dollar sign.
* Class name cannot use reserved words or keywords.
* Class name can’t have space between words.

**1.6 Keywords in java**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| abstract | char | else | goto | long | return | this | while |
| assert | class | extends | if | native | short | throw |  |
| boolean | const | enum | implements | new | static | throws |  |
| break | continue | final | import | package | strictfp | transient |  |
| byte | do | finally | instanceof | private | super | try |  |
| case | default | float | int | protected | switch | void |  |
| catch | double | for | interface | public | synchronized | volatile |  |