JRE

- JRE stands for Java Runtime Environment.
- It provides an environment at runtime for running java program
- It contains a set of supporting libraries.
- It provides supports with classes and various other files that are used by JVM.
- It is the implementation of JVM.
- The JRE consists of various components, these are given below..
    - Abstract Window Toolkit (AWT), Swing, Image Input / Output, Accessibility, drag, and drop, etc.
    - Java plug-in.
    - Libraries for Input/Output, extension mechanisms, beans, JMX, JNI, networking, override mechanisms, etc.
    - Java Database Connectivity (JDBC), Java Naming and Directory Interface (JNDI), Remote Method Invocation (RMI).
    - Lang and util base libraries which include lang and util etc.

And many more components.

JDK

- The java Environment comes with a collection of tools and thousands of classes and methods.
- The development tools are the part of the system known as Java development Kit.
- The classes and methods are the parts of java standard library(JSL) also known as Application programming interface(API).
- The JDK comes with a collection of tools that are used for developing and running java program.
- It basic tools are..
    - applet Viewer

      appletviewer run and debug applets without a web browser, its standalone command-line program to run Java applets.
    - Jar

The jar is a package file format that contains class, text, images and sound files for a Java application or applet gathered into a single compressed file.

- Javac

  javac is the compiler for the Java programming language; it's used to compile .java file. It creates a class file which can be run by using java command.

- Java

  It is an interpreter. When a class file has been created, the java command can be used to run the Java program.

- Javap

  The javap command is known as the Java Disassembler. The javap command disassembles one or more class files.

- Javah

  javah produces C header files and C source files from a Java class

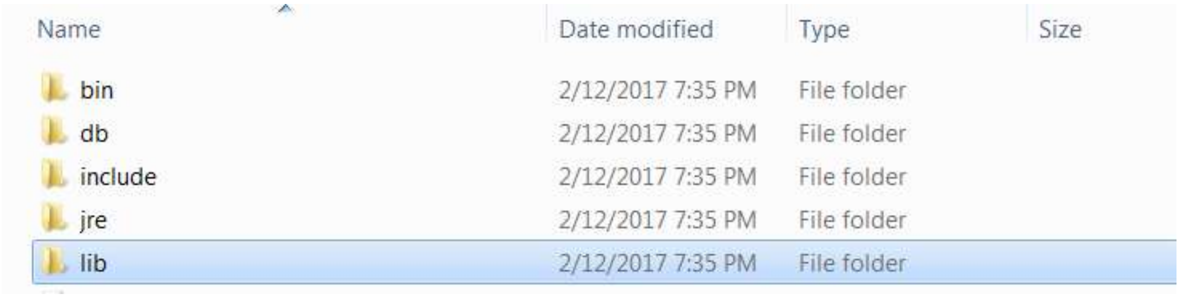
- Javadoc

  The javadoc command parses the declarations and documentation comments in a set of Java source files and produces a corresponding set of HTML page.

Sub folder created by JDK

After installing JDK several related folder created on hardware location. The location of this folder depends upon particular system but generally the JDK folder formed under program files/java/javaversion

The following folder created in JDK root folder…



Name	Date modified	Type	Size
bin	2/12/2017 7:35 PM	File folder	
db	2/12/2017 7:35 PM	File folder	
include	2/12/2017 7:35 PM	File folder	
jre	2/12/2017 7:35 PM	File folder	
lib	2/12/2017 7:35 PM	File folder	

bin :- The compiler and other java development tools are stored.
db :- Java Db is installed in the db directory of the JDK installation. This distribution contains scripts and libraries.
Include :- It contains files to integrate java with other language.
Jre :- Runtime Environment Files are contained.
Lib :- Library Files including API libraries.

API

API stands for application programming interface.
The java standard library includes hundreds of classes and methods group into several functional packages .most commonly used packages are:-

1. **language support package**
   A collection of classes and methods required for implementing basic features of java to use their functions and classes we import it.
   import java.lang;

2. **Utility package**
   A collection of classes to provide utility function such as delete and time function to use them re-import util package.
   import java.util;

3. **Input output package**
   A collection of classes and methods required for i/o manipulations we use them by importing i/o package.
   import java.io;

4. **Networking package**
   A collection of classes and methods required for communication with other classes.
   import java.net;
5. **Awt package**
   Awt abstract window toolkit contains classes that implement platform independent GUI . To use it we write. . .
   import java.awt;

6. **applet**
   This includes a set of classes that allow us to create java applet.

Import java.applet.*;

Java program structure

A java program may contain one or more section which is shown below:-

```
Documentation section
 package system
 import system
 interface statement
 Class defination
       Main method class
       {
          main method definition;
       }
```

1. **Documentation section**

   Here we can use a set of comments line to describe about program or programer.the java supports three type of comments which will discuss later but we can use comment line agrument here in our program becoude compiler always ignore it.

2. **package statement**

   The java treated the use of package statement.it is the first statement in java. This statement declares a package name and inform the compiler that the classes defined here belong to this package.

   ex: package student

   The package statement is optional ie. Our class does not have to be part of a package.

3. **import statement**

   In this statement we import some statement.this is similar to the #include statement in c.

   ex: import student.test

   This statement instructs the interpreter to load the test class contain in the package student. Using import statement, we can have access to classes that are part of other named package. We use import statement when we are in need.

4. **interface statement**

   An interface is like a class but includes a group of method declaration. This is only when we wish to implement the multiple inheritance feature in the program.

   Note: more about interface is discussed later.

5. **class definition**

   We know that java is pure object oriented programming language,every function defines under class. So a program may contain several classes. Classes are the primary and essential elements of java program. These classes are used to solve the real world problems.

6. **main method class**

   Since every java standalone requires main method for the execution. we have already discussed that every method must be the part of a class .here main is also method then we must include the main method class for run a java standalone program.

How to enter java program/precautions during typing program

Sun java does not have an integrated development environment like the IDE of vc++, vb, c, ++ etc. A java program is written on any editor like dos editor, notepad, wordpad or any other word processor, but java program must have the ".java" extension.

To make an error/bug free program we must consider about java syntax and grammar. Java is case sensitive so we must careful during program coding we must keep in mind that which letter capital and which will will small.

Saving a java program

1. **file name and extension**

   In every language, file plays a great role to run a program. In java we also give a file name but this file name is must as same as the class name in which main is defined. We know that a java application may have more than a one class but we give that class name as file name which keep definition of main.
   With the filename we must give the extension '.java' because sun java has not its own IDE so the default extension is depend upon editor. (It means that by default it cannot take ".java" extension).

(a) During saving the program if we do not give the file name as class name then the compiler will generate an error during compile time.
i.e. Exception in thread "main language".
No class found error: file name.

(b) If more than one class is present in program and saving time we are not giving the file name as the same class in which main is defined but we are giving other which is present as class in program then compiler will compile without any error but runtime the interpreter will produce an error .
Exception in thread 'main' java.lang.no such method error.
So we always give the file name as same as the class name in which main is defined.

2. **Location where file is to be save**

To compile or run a program it is necessary to save the program where the java compiler (java) and java interpreter (java) is present .generally it is present in the bin sub directory of jdk version (or java). So we should save our program in bin directory. But we can save our program also in specific desired folder.

Compiling and running a java standalone application

We will see the compilation and running process in two case :-

1. When program is in bin directory.
2. When program is in desired directory.

In both cases to run a java program we use the dos prompt.let us see in first case:

1. When program is in bin directory

To compile

We type on dos prompt   javac filename.java

Here javac is java's compiler and filename is class name in which main is defined. If there is no error then next dos prompt appear in front of your screen, then we think that the program has been compiled successfully.

To run

We type on dos prompt   java filename here java is interpreter and filename is that which is compiled successfully. If there is no bug/error then we see the desired output.

2. when program is in desired directory

Here the compilation and running process is same but we will have to set the path of jdk where, java compiler (javac) java interpreter (java) and other jdk tools are present .we set this path through path command.

Here firstly we have to go on that directory where java program is present through dos command. Let us see the following steps:

- Open the dos prompt.

- Then select the desired folder

```
f:\>cd folder name (enter)
f:\\cd amar_java(enter)
```

Then copy the path where javac and other jdk tools are present and then set the path through pasting it on dos prompt with path command.

Ie. F:\ amar_java>path=paste path(enter)

Now we are on stage where we compile and run our program.

To compile

Javac file_name.java(enter)

To run

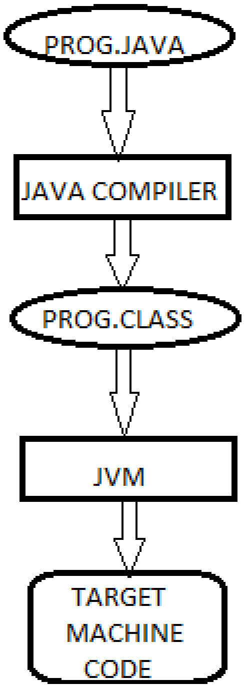
Java file_name(enter)

What happen when we compile and run java program

When we compile a program through javac compiler then it produce class file with '.class' extension. It produce class file for every class present in program not only that in which main define.

These class file are known as byte code .byte code is not the direct executable code. It is only understandable by jvm. It is platform independent because it is not the machine code. We can run this byte code in any machine where jvm is present. The jvm reads this byte code and interpreter interpret it into target machine code.

The diagrammatic presentation of execution of java program is shown in following figure. The figure shows how the core java program or java source code moves through several phases during execution.



My first program in java

```
class First
```

```
{
        public static void main(String args[])
         {
                System.Out.println("welcome to java program");
         }
}
```

This program is used to display "welcome to java program”. It shows how we create main method class and what is the work of method main? How main looks? All things are done in class. Let’s look this program line by line.

class First

Here this line indicates that we are creating a new class First. Here class is a keyword. It is used to declare that the new class, "first" is being defined. First is an identifier that is name of the class. After we translate this class into byte code
de, the java virtual machine will be able to create object/ instance of this class and run them.

{....................} -- Every class definition in java begins with an opening brace "{" and ends with a matching "}" closing braces. It enclosed blocks of code within these curly braces.

It means member definition and entire class definition take place between these braces.

The code inside the block has its own scope or visibility.

public static void main(String args[])

It defines the method main. Every java standalone application must have main method . Because, the execution starts from main method. So the JVM will look for when starts execution, when it finds the main method the JVM passes control to it.

The main method must be declared with the public access specifier. It declares the main method as unprotected and therefore making it accessible to all other classes or we can say that it may be called outside its classes.

The main method must be declared with keyword static. It states that method as on that belongs to the entire class and not a part of any object of the class. The main must be declared as static since the interpreter was this method before any objects are created.

The keyword static allows main() to be called without having the instance of class(generally we know that member function [method]) calls with the help of the object but here main method call without any object creation . So, we can say that main is a class method not an object method.

The void keyword shows the return type of method main. Here void shows that main will not return a value.

Let us see what is in the parenthesis of main().any information that is needed to a method, is received by variable specified within the set of parenthesis , these variables are called arguments.

String args[] declare a parameter named args which is array instance of class Strings. Here parameter is indicating that main is receiving an array of String value called arg.

These string values hold the value passed from command line when we run the application.

Ex: we type, **java first** hi Amar then "hi" and "Amar" would be the two strings in the args array.