**Java jar, war and ear files:**

A jar (Java Archive) is a package file format typically used to aggregate many Java .class files and associated metadata and resources (text file, image file, etc.) into one file to distribute application software or libraries on java platform.

**To create a jar file:[ c – create v – verbose f- filename].**

* jar -cvf <jarfilename.jar> <filename.class> --- with one class file.
* Jar -cvf <jarfilename.jar> A.class B.class C.class -- with three class files.
* Jar -cvf <jarfilename.jar> \*.class -- with all class files.
* Jar -cvf <jarfilename.jar \*.\* -- with all files in current working directory.

**To run a jar file:**

* Java -jar <jarfilename.jar> -- to run a jar file.

**To extract a jar file:**

* Jar -xvf <jarfilename.jar> -- extracts jar file.

**To view table of contents:**

* Jar -tvf <jarfilename.jar> -- to view table of contents.

**To update a jar file:**

* Jar -uf <jarfilename.jar> inputfile(s) -- to update jar file with new files.

Note: When we create a jar file, it automatically receives the default manifest file, there can be only one manifest file and it always has the classpath. (META-INF/MANIFEST.MF).

**META-INF/MANIFEST.MF**  : contains information about other files which are packed.

[Note: To use jar file we need to mention .jar filename]

**Note: Same commands are used to create .war file and .ear file with corresponding extension.**

For **.war** file we have special jar file with **web.xml file** in the **WEB-INF folder.**

For **.ear** file we have special jar file containing **application.xml file** in the **META-INF folder**

**war file:**  contains web applications ( jsp, servlets, ejb, etc) each war file represents an web application.

**ear file**: contains anything from J2EE, represents ( jsp, servlets, ejb, etc).

**java:** console output will be executed, means sop will be displayed but other programs will run.

**javaw:** console output will not be executed, means no sop, uses esp, GUI based execution.

[ No associated console window, Java-window].

**javaws:** used to download and run the distributed web application, no console is associated.

[ java web start utility].

[Note: all are part of JRE and use same JVM].

**classpath:** describes the location of where required .class files are available.

**path:** represents location where binary executable are available.

[ Java compiler and JVM will use them].

**bin:** binary executables

**Java Binary:** Java source file is compiled into binary class file. These binary files are used in java virtual machine for execution.

**Web Server:** provides support only for web applications ( eg. Servlets, jsp, html etc.).

**Application Server:** any technology java J2EE (eg. Servlets, jsp, html, EJB, JMS etc.).

[ Note: in Application server there is built in Web Server].

**strictfp:** is a keyword in java, used for restricting floating point calculations and ensuring same result on every platform operation in floating point variable.

**volatile:** is a keyword in java, using volatile is making class **thread safe**.

**Thread safe:** means that a method or class instance can be used by multiple threads at the same time without any problem.

**Factory Method:** one java class method is able to return same class object or different class object.