What is build process?
    Keeping resources ready=> Arranging them in various folders=> add libraries/jar to classpath => compilation => execution=> testing
    => packing for release or deployment is continuously required in project development and delivery process.

Core Java Project build process
==========
 a. develop java resources and other files
 b. keep them in different folders
 c. add jar files to classpath
 d. compilation
 e. execution/testing[performing testing on his own piece of code is called "UnitTesting"]
 f. packing the app for release.

Performing build process activities manually is having lots of limitations
      a. remembering compilated and repeatitive operations is very tough.
      b. we may mismatch order.
      c. we may forget certain activities
      d. doing multiple activities of build process manually will waste the time.

To automate this process activities we can use .bat file
==========
   run.bat
   =======
      cd e:
      md xyz
      cd xyz
      copy ... ...
      copy ... ...
      set path=...
      set classpath = ...
      javac -d *.java
      java <pkg>.<MainClass>

   cmd> run.bat

batch file is given to combine all related commands into single command[by using single command we can automate the process]

limitations of batch files
==========
 a. Conditional execution is not possible.
 b. we can not create dependancy among the operation.
 c. jar files must be added dynamically no dynamic downloading of jar file from internet
 d. if one command files is batch file .next command will not execute.
 e. It is not declarative[not self intelligent ie, we need to tell everything to do]

To overcome some of these problems we got ant tool[Another Neat Tool]
  a. It is same as batch files, but we can keep operations as conditional operations and we can create dependency among
       the operations.

To overcome both these tools problem we got "Maven" tools with lots of advanced features

=> Maven is just not a build tool it is also called as "Project Management tool".

KeyFeatures of maven
====================
1. Maven tries to avoid so much configurations as possible by chooosing real default values and supplying
   project templates[archetypes]
2. Can download jars automatically.
3. Can maintatin mulitple repositories having jar files, plugins etc
4. Provides standard project directory structure.
5. Gives maven inheritance to share jar files and plugin among the multiple projects.
6. Allows to develop multi module projects.
7. Can generate the war,jar,ear file based on the application componenets.
8. Can generate the project documentation.
9. Can run unit tests and can generate unit test reports.
10. Can clean and install the projects in the local servers or remote servers.

Archetypes[Project templates]
=============================
 1. maven-archetype-quickstart[for standalone projects]
 2. maven-archetype-webapp[for webapplications]

Note: archetypes are project directory structure models

Maven can be build in 2 ways
    a. In command line mode
    b. From IDE Like Eclipse,Intelij,Netbeans,Eclipselink,....

To keep maven in our system
===================================
a. Download zip file and extract it from the following link
          https://maven.apache.org/download.cgi(send one in the link => apache-maven-3.9.1-bin.zip)
b. Create the following environment variables
      a. Add <maven_home>\bin to path environment variables.
                    set path=D:\jars\apache-maven-3.9.1\bin
      b. Add java installation folder to JAVA_HOME environment variables.
                  set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_202

To check whether maven is installed properly or not just type mvn -version in command prompt
D:\>mvn -version
Apache Maven 3.9.1 (2e178502fcdbffc201671fb2537d0cb4b4cc58f8)
Maven home: D:\jars\apache-maven-3.9.1
Java version: 1.8.0_202, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_202\jre
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

Maven repositories
==================
 => Repositories is a small db or folder which holds items.
 => Maven repositories can hold
      a. jar files/libraries/dependancies
        b. plugins(patch software to provide additional functionalities)
      c. old/sample projects[Springboot==> Old project==> inherited through => Maveninheritance]

In Maven everything (jar/plugin/project) is identified with 3 details[GAV]

a. artifactId(jar file name/plugin name/projectname)
b. groupId(company name)
c. version(jar/plugin/project version)[stable(currently in use),SNAPSHOT(next version not stable),RELEASE(next version ready)]

eg#1.
GroupId   => pivotal team
artifactId => spring-aspects
version   => 5.3.17

Repositories
============
a. Central Repositories(given by maven people)
Central Repository:
Avaialble in internet, managed by Apache Maven Community. When Maven does not find any dependency in local repository, it starts searching in central repository.
URL for central repository: http://repo.maven.apache.org/maven2
Generally maintains free jars and plugins of open source technologies

b. Local Repositories(In every machine where maven is required)
It is user specific repository, generally it will be collected from TL/PL who creates Maven project directory structure.
Contains jars, plugins, current project related packings and etc..
Default location: C:\users\<usernmae>\.m2
Will be created automatically for any maven command apart from (mvn -version)
Location can be changed through <maven_home>\conf\settings.xml file using
<localRepository>d:\\maven</localRepository>

c. Remote Repositories(Give by third party companies)

Creating an Project using MAVEN in CLI Mode
==
1. Open the Command Prompt and change directory where you want to create your project and call
mvn archetype:generate (goal to begin the process).

D:\maven>mvn archetype:generate(press enter key)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 2036:
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8:
Define value for property 'groupId': ineuron
Define value for property 'artifactId': MathProj1
Define value for property 'version' 1.0-SNAPSHOT: : 1.0
Define value for property 'package' ineuron: : in.ineuron
Confirm properties configuration:
groupId: ineuron
artifactId: MathProj1
version: 1.0

```
package: in.ineuron
Y: :
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: maven-
archetype-quickstart:1.4
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: ineuron
[INFO] Parameter: artifactId, Value: MathProj
[INFO] Parameter: version, Value: 1.0
[INFO] Parameter: package, Value: in.ineuron
[INFO] Parameter: packageInPathFormat, Value: in/ineuron
[INFO] Parameter: package, Value: in.ineuron
[INFO] Parameter: version, Value: 1.0
[INFO] Parameter: groupId, Value: ineuron
[INFO] Parameter: artifactId, Value: MathProj
[INFO] Project created from Archetype in dir: D:\Mavenpgms\MathProj1
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  05:18 min
[INFO] Finished at: 2023-04-03T12:35:58+05:30
[INFO] ------------------------------------------------------------------------


MathProj1
   |=> src/main/java[source code]
                |=> in.ineuron
                              |=> MathApp.java(source code)
   |=> src/test/java[test code-> unit testing]
                  |=> in.ineuron
                              |=> AppTest.java(test code)
   |=> pom.xml(build file)===> groupId,artifactId,version


MathApp.java
============
package in.ineuron;
public class MathApp {
      public int add(int x, int y){
            return x+y;
      }
    public static void main( String[] args ){
        MathApp m = new MathApp();
            int result = m.add(10,20);
            System.out.println("The sum is :: "+result);
    }
}

In the command prompt execute the following life cycle actions
a. mvn package
      >» Generates jar files in target folder having <projectname>-ver. jar file
b. mvn clean
      »  Cleans the project .. deletes target folder
c. mvn clean package
      » Cleans the project and also creates jar file with latest code
d. mvn compile
      >> compile the project code and generate the .class file in target folder.

To run jar file App manually
```

```
      D:\mavenpgms\MathProjl>java -cp target/MathProj1-1.0.jar
in.ineuron.Arithmetic

The Maven Life cycles are
a. clean(3 phases)
b. default(23 phases)
c. site (4 phases)

=> Each life cycle of maven will have lot of phases.
=> These phases are already linked with plugins to peform certain operations, but
we can configure extra plugins to perform
   more operations.

To test our code writing unit test case through Junit
=====================================================
package in.ineuron;
import junit.framework.*;
import in.ineuron.*;
public class AppTest extends TestCase{
      public void testSumWithPositiveNumber(){
        MathApp ar = new MathApp();
            int actual = ar.add(10,20);
            int expected = 30;
            assertEquals(actual,expected);
    }
      public void testSumWithNegativeNumber(){
        MathApp ar = new MathApp();
            int actual = ar.add(-10,-20);
            int expected = -30;
            assertEquals(actual,expected);
    }
      public void testSumWithMixedNumber(){
        MathApp ar = new MathApp();
            int actual = ar.add(-10,20);
            int expected = 10;
            assertEquals(actual,expected);
    }
      public void testSumWithZero(){
        MathApp ar = new MathApp();
            int actual = ar.add(0,0);
            int expected = 0;
            assertEquals(actual,expected);
    }
}
D:\Mavenpgms\MathProj1>mvn test
                                    => runs all the test cases and generates the
report in command line.
D:\Mavenpgms\MathProj1>mvn surefire-report:report
                                    => go to target folder search for a file called
surefire-report.html
D:\Mavenpgms\MathProj1>mvn site
                                    => go to target folder search for a file called
index.html

Maven can't execute the java app directly becoz there is no life cycle phases for
that.
 To use that we need to use an extra plugin called :: exec-maven-plugin
<build>
    <plugins>
```

```
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>3.1.0</version>
        <executions>
          <execution>
                  <id>ArithmeticApp</id>
                  <phase>package</phase>
            <goals>
              <goal>java</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <mainClass>in.ineuron.MathApp</mainClass>
        </configuration>
      </plugin>
    </plugins>
</build>
```

```
D:\Mavenpgms\MathProj1>mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] -------------------------< ineuron:MathProj1 >-------------------------
[INFO] Building MathProj1 1.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ MathProj1 ---
[INFO] Deleting D:\Mavenpgms\MathProj1\target
[INFO]
[INFO] --- resources:3.3.0:resources (default-resources) @ MathProj1 ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\Mavenpgms\MathProj1\src\main\
resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ MathProj1 ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e.
build is platform dependent!
[INFO] Compiling 1 source file to D:\Mavenpgms\MathProj1\target\classes
[INFO]
[INFO] --- resources:3.3.0:testResources (default-testResources) @ MathProj1 ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\Mavenpgms\MathProj1\src\test\
resources
[INFO]
[INFO] --- compiler:3.10.1:testCompile (default-testCompile) @ MathProj1 ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e.
build is platform dependent!
[INFO] Compiling 1 source file to D:\Mavenpgms\MathProj1\target\test-classes
[INFO]
[INFO] --- surefire:3.0.0:test (default-test) @ MathProj1 ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit.JUnit3Provider
[INFO]
```

```
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running in.ineuron.AppTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.011 s - in
in.ineuron.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ MathProj1 ---
[INFO] Building jar: D:\Mavenpgms\MathProj1\target\MathProj1-1.0.jar
[INFO]
[INFO] --- exec:3.1.0:java (ArithmeticOperation) @ MathProj1 ---
The sum is :: 300
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.162 s
[INFO] Finished at: 2023-04-04T22:59:06+05:30
[INFO] ------------------------------------------------------------------------

D:\Mavenpgms\MathProj1>java -cp target\MathProj1-1.0.jar  in.ineuron.MathApp
The sum is :: 300
```