Task 1. working and the differences between Maven, Gradle and SBT .

1) **Maven**:

Maven is a "build management tool", it is for defining how your .java files get compiled to .

class, packaged into .jar (or .war or .ear) files,

processed with tools, managing your class path, and all others sorts of tasks that are required to build your project.

It is similar to Apache Ant or Gradle or Make files in C/C++,

but it attempts to be completely self-contained in it that you shouldn't need any additional tools or scripts by

incorporating other common tasks like downloading & installing

necessary libraries etc. It is also designed around the "build portability" theme, so that you don't get issues as

having the same code with the same build script working on one computer

but not on another one.

Gradle:

Gradle is another build system that takes the best features from other build systems and combines them into one.

It is improved based off of their shortcomings.

It is a JVM based build system, what that means is that you can write your own script in Java, which Android Studio makes use of.

One cool thing about gradle is that it is a plugin based system. This means if you have your own programming language

and you want to automate the task of building

some package (output like a JAR for Java) from sources then you can write a complete plugin in Java or Groovy,

and distribute it to rest of world.

SBT:

SBT is an open source build tool for Scala and Java projects, similar to Java's Maven or Ant.

Its main features are:

1) Native support for compiling Scala code and integrating with many Scala test frameworks

2) Build descriptions written in Scala using a DSL

3) Dependency management using Ivy (which supports Maven-format repositories)

4) Continuous compilation, testing, and deployment

5) Integration with the Scala interpreter for rapid iteration and debugging

6) Support for mixed Java/Scala projects

DIFFERENCES:

SBT is very simple and it is focused on Scala it relies on Ivy for dependency management.

Maven it's a great build tool and it enables to control the entire software lifecycle with XML files.

Using the Project Object Model you can intercept all points of the software lifecycle from compile to test,

packaging and deploy.

Maven has it's own dependency manager. The bad issue in Maven is the XML syntax, writing a POM can be annoying and too much expensive.

On the other hand Gradle is built on top of Maven, Ant and Ivy. It uses Maven repositories. Gradle doesn't use XML,

it's a polyglot build tool.

It combines the Ant API with the Groovy language to enable developers to write a build script with an intuitive DSL.

With a few lines of code

you can write a Gradle build script that can do the same things that Maven can do. With Gradle you can define your own

task with the Groovy language

and intercept programmatically your build execution. This functional approach is not for all developers, infact Maven it's

good if you don't want this

behavior in your build environment. Both Maven and Gradle have plugins to integrate your build with technologies used in your projects.