Lecture 0 - Course Introduction

Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

CSC-1004: Computational Laboratory Using Java
Course Page: [Click]

CSC 1004 Resources

Please visit our course page.

SCAN ME

Please click this link.

Please post your question on the discussion board in the BlackBoard (BB) system.

- Step 1: Search for existing questions.
- Step 2: Create a thread.
- Step 3: Post your question.

Please click this link.

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Course Introduction

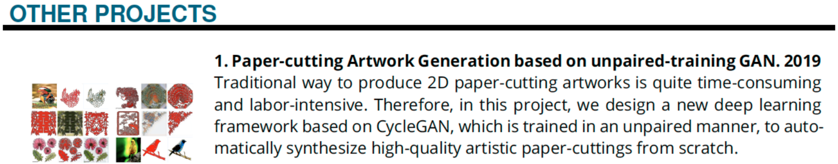
This course is a computational lab course to strengthen programming skills. As a laboratory course, CSC-1004 will be delivered in the format of finishing projects. Self-learning and debugging skills are especially important for succeeding in this course. Each student is expected to implement individually:

- Two medium-sized Java projects (around 2000 lines of code).
- A smaller Python project (around 500 lines of code).

Course Introduction

Goals: To strengthen programming skills:

- *Prepare for other CS Courses.* E.g., Data Structure, Machine Learning, etc.
- *Collect Project Experience.* E.g., the project could appear on your resume or CV for finding an internship.



OTHER PROJECTS

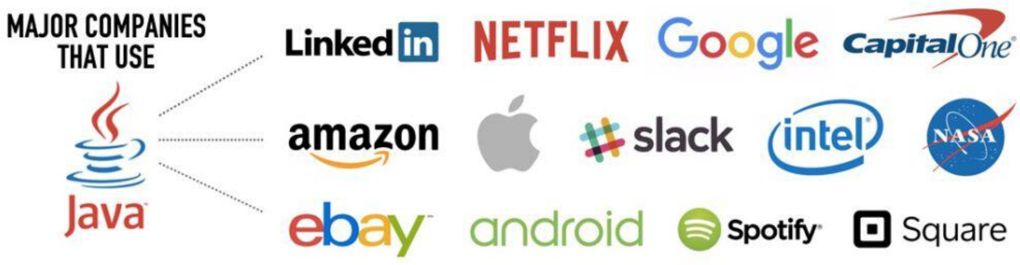
1. Paper-cutting Artwork Generation based on unpaired-training GAN. 2019
Traditional way to produce 2D paper-cutting artworks is quite time-consuming and labor-intensive. Therefore, in this project, we design a new deep learning framework based on CycleGAN, which is trained in an unpaired manner, to automatically synthesize high-quality artistic paper-cuttings from scratch.

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Course Introduction

"Why we need to learn Java programming"

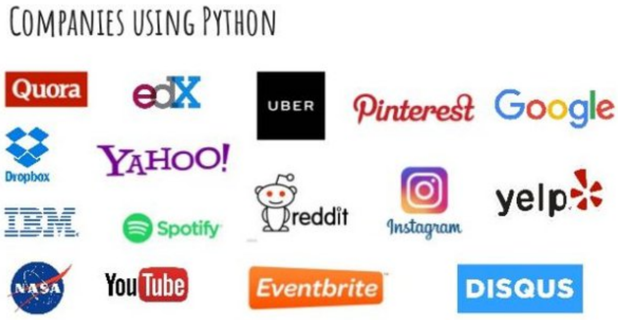
Java has been one of the *most popular programming languages for many years*. It's used in a variety of applications, from web development to mobile apps (especially Android), desktop applications, and large-scale enterprise systems.


MAJOR COMPANIES THAT USE
Java
LinkedIn
NETFLIX
Google
Capital One
amazon
slack
intel
NASA
ebay
android
Spotify
Square

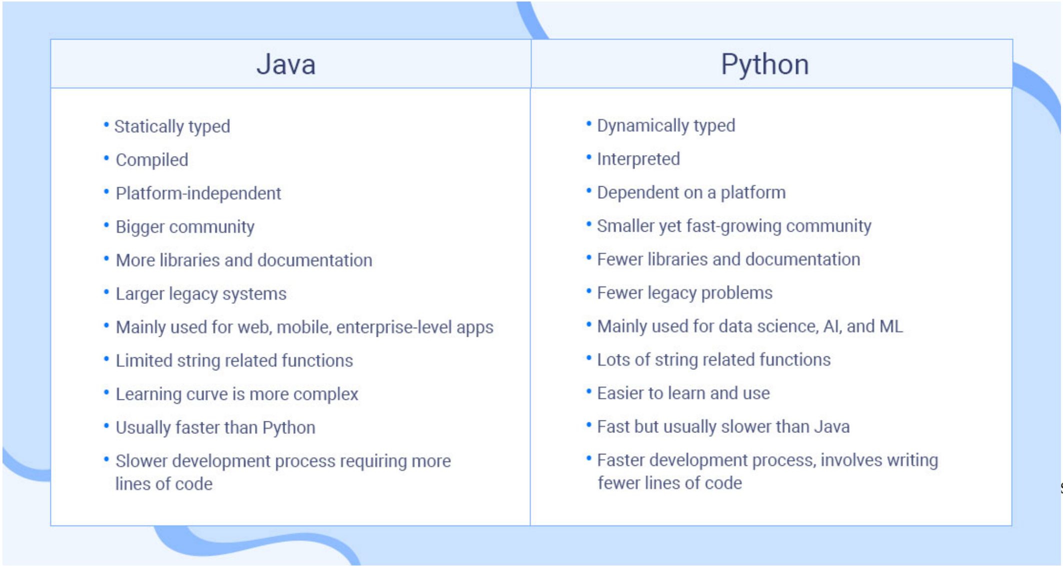
Course Introduction

"Why we need to learn Python programming"

There's a strong market demand for Python developers in various domains, such as web development, data science, artificial intelligence, and more. Python skills are often sought after by employers.


COMPANIES USING PYTHON
Quora
edX
UBER
Pinterest
Google
Dropbox
YAHOO!
reddit
Instagram
yelp
IBM
Spotify
NASA
YouTube
Eventbrite
DISQUS

Course Introduction



Java	Python
Statically typed	Dynamically typed
Compiled	Interpreted
Platform-independent	Dependent on a platform
Bigger community	Smaller yet fast-growing community
More libraries and documentation	Fewer libraries and documentation
Larger legacy systems	Fewer legacy problems
Mainly used for web, mobile, enterprise-level apps	Mainly used for data science, AI, and ML
Limited string related functions	Lots of string related functions
Learning curve is more complex	Easier to learn and use
Usually faster than Python	Fast but usually slower than Java
Slower development process requiring more lines of code	Faster development process, involves writing fewer lines of code

Course Introduction

Jobs that need Java and Python programming skills:

- **Software Engineer.** Proficiency in languages like C/C++, Java and Python. (Database, Data Structures, and Web ...) Salary:
    - China: CNY 200,000 to CNY 400,000 per year.
    - Europe: EUR 50,000 - EUR 70,000 per year.
    - USA: USD 80,000 - USD 120,000 per year.
- **Data Analysts.** Knowledge of languages like Python and R (a language for statistical computing). Salary:
    - China: CNY 60,000 to CNY 200,000 per year.
    - Europe: EUR 40,000 to EUR 70,000 per year.
    - USA: USD 60,000 to USD 90,000 per year.

Course Introduction

Jobs that need Java and Python programming skills:

- **Applied Scientist.** Strong working knowledge of programming languages such as C/C++, Java, or Python. Practical machine learning experience. Salary:
  - China: CNY 200,000 to CNY 500,000 per year.
  - Europe: EUR 50,000 - EUR 80,000 per year.
  - USA: USD 80,000 - USD 150,000 per year.
- **Research Scientist.** Strong working knowledge of programming languages such as Python, Java, R or Matlab. Statistical and Mathematical Modeling. Databases,. etc. Salary:
  - China: CNY 180,000 to CNY 500,000 per year.
  - Europe: EUR 50,000 - EUR 80,000 per year.
  - USA: USD 70,000 - USD 120,000 per year.

Course Introduction

CSC 1003 v.s., CSC 1004

- Goal: Basic knowledge of computer science and Java programming.
- Approach: Remember conceptions, implement features, and pass exams.
- Evaluation: assignments and exams.

- Goal: Build a project (something useful) with your knowledge.
- Approach: Practical training that makes you a qualified programmer.
- Evaluation: Let's coding!

Course Introduction

Scoring Schema:

- *Java Projects (50% in total, each 25%).* Basic functions and videos.
- *Python Project (25%).* Basic functions and videos.
- *Advanced Features (25%).* Implement two advanced features (see intro later).

Course Introduction

1. Java Chat Zoom (Check this page.) You are expected to write a chat room with Java. The main features include:

1. Java Class and Inheritance.
2. Java Multi-Threading.
3. Java Socket Programming.

Course Introduction

2. Java Game (Check this page.) You are expected to write a game named Gomoku ("Five In A Row") with Java. The basic features include:

1. Java Class and Inheritance.
2. Java Graphical User Interface (GUI).

Course Introduction

3. Python Games (Check this page.) You are expected to write a game with Python. The basic features include:

1. Python Package Usage.
2. Python Class and Inheritance.
3. Python Graphical User Interface (GUI).

What to Expect

What to Expect:

- *Instructor and TAs will introduce the core techniques for programming the project.*
- *Instructor and TAs will introduce general requirements, leaving some space for including additional designs.*
- *Every student must individually finish the projects* **without forming any team.**
- *Students must learn how to design the projects and debug the codes by themselves.*

Self-learning skills are extremely important for success in CSC 1004.

Lectures and Tutorials

Lectures:

- *Introduce the programming skills to complete the basic features in your project.*
- *Introduce some advanced Java/python knowledge to improve your project.*

The lecture covers the main idea, how to utilize these ideas in your project is your work.

Tutorials:

- *Go over the code of some example projects with you.*
- *Question answering.*

TAs will not debug for you or write any code for you.

Java IDEs

Popular **Integrated Development Environment (IDE)** for Java includes:

- *Eclipse (Not Recommended!).* (Official Site) It is a Java-based open-source platform that enables us to create highly customized IDEs from Eclipse members' plug-in components.

Java IDEs

Popular **Integrated Development Environment (IDE)** for Java includes:

- *Eclipse (Not Recommended!).* (Official Site)
- *NetBeans (Not Recommended!).* (Official Site) NetBeans is a Java-based IDE and basic application platform framework. Besides Java, JavaScript and JavaFX, NetBeans supports PHP, C/C++, Groovy, and HTML5 languages.

Apache
NetBeans IDE

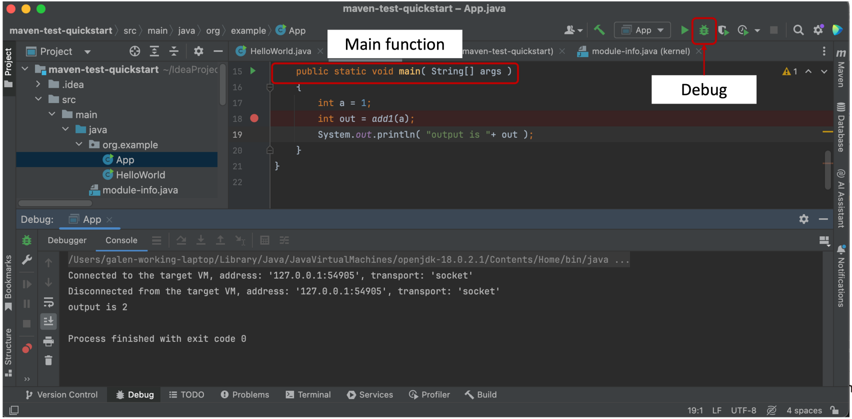
Java IDEs

Popular **Integrated Development Environment (IDE)** for Java includes:

- *Eclipse (Not Recommended!).* (Official Site)
- *NetBeans (Not Recommended!).* (Official Site)
- *IntelliJ IDEA (Recommended!).* (Official Site) It is a free and open-source commercial Java IDE. It has several features to make programming easier, like code completion, **debugging**, XML editing support, code refactoring, code checks, unit testing, etc.

Debugging with IntelliJ IDEA.

- Download the **Ultimate** version of IntelliJ IDEA (register with your school's email). Find the main file and open the debug mode.


maven-test-quickstart – App.java
maven-test-quickstart 〉 src 〉 main 〉 java 〉 org 〉 example 〉 App
App
Project
maven-test-quickstart ~/IdeaProjec
.idea
src
main
java
org.example
App
HelloWorld
module-info.java
HelloWorld.java
maven-test-quickstart)
module-info.java (kernel)
Maven
Database
AI Assistant
Notifications
Bookmarks
Structure
Debug:
Debugger
Console
Version Control
Debug
TODO
Problems
Terminal
Services
Profiler
Build
19:1 LF UTF-8 4 spaces

Main function

Debug

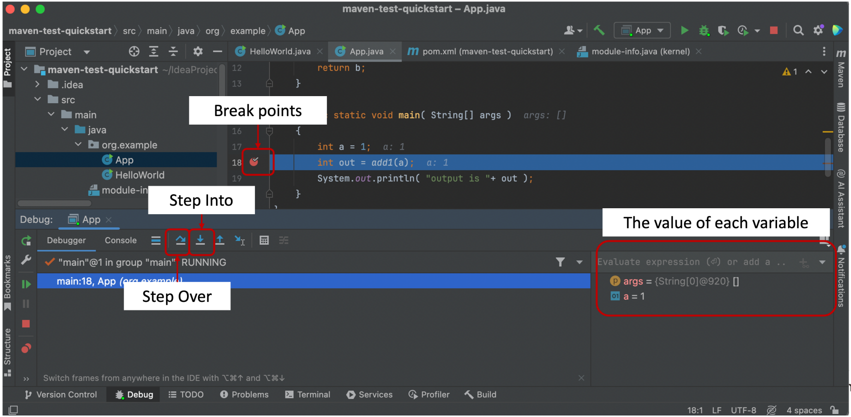
```
    public static void main( String[] args )
    {
        int a = 1;
        int out = add1(a);
        System.out.println( "output is "+ out );
    }
}
```

```
/Users/galen-working-laptop/Library/Java/JavaVirtualMachines/openjdk-18.0.2.1/Contents/Home/bin/java ...
Connected to the target VM, address: '127.0.0.1:54905', transport: 'socket'
Disconnected from the target VM, address: '127.0.0.1:54905', transport: 'socket'
output is 2

Process finished with exit code 0
```

Debugging with IntelliJ IDEA.

- **Breakpoints** allows stopping program execution at a certain point. Breakpoints are set by hovering the mouse over the gutter area and clicking on it.



Break points

Step Into

Step Over

The value of each variable

```
        return b;
    }

    static void main( String[] args )   args: []
    {
        int a = 1;   a: 1
        int out = add1(a);   a: 1
        System.out.println( "output is "+ out );
    }
```

"main"@1 in group "main": RUNNING

main:18, App (org.example)

args = {String[0]@920} []

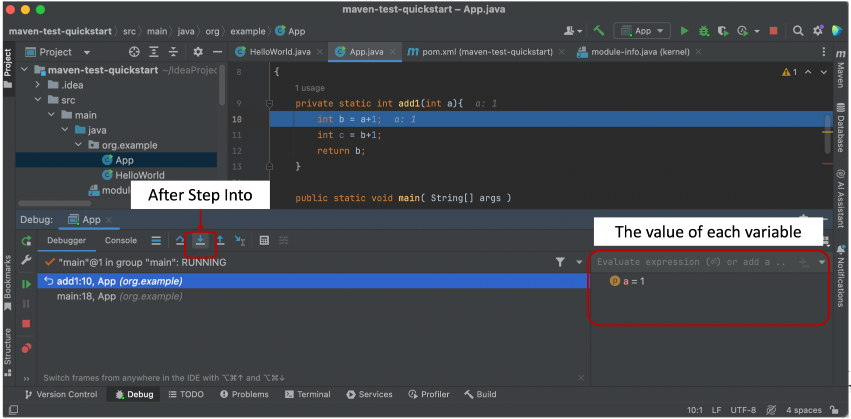
a = 1

Switch frames from anywhere in the IDE with ⌥⌘↑ and ⌥⌘↓

maven-test-quickstart – App.java
maven-test-quickstart 〉 src 〉 main 〉 java 〉 org 〉 example 〉 App
Project
maven-test-quickstart ~/IdeaProjec
.idea
src
main
java
org.example
App
HelloWorld
module-in
HelloWorld.java
App.java
pom.xml (maven-test-quickstart)
module-info.java (kernel)
Debug:
App
Debugger
Console
Evaluate expression (⏎) or add a ..
Version Control
Debug
TODO
Problems
Terminal
Services
Profiler
Build
18:1
LF
UTF-8
4 spaces
Maven
Database
AI Assistant
Notifications
Bookmarks
Structure

Debugging with IntelliJ IDEA.

- **Step into**. While debugging, if a function is encountered and a step into action is selected, the debugger will stop program execution at each point of that function.


maven-test-quickstart – App.java
maven-test-quickstart 〉 src 〉 main 〉 java 〉 org 〉 example 〉 App
Project
maven-test-quickstart ~/IdeaProjec
.idea
src
main
java
org.example
App
HelloWorld
modul
HelloWorld.java
App.java
pom.xml (maven-test-quickstart)
module-info.java (kernel)
Debug:
App
Debugger
Console
Version Control
Debug
TODO
Problems
Terminal
Services
Profiler
Build
10:1 LF UTF-8 4 spaces
Maven
Database
AI Assistant
Notifications
Bookmarks
Structure
Evaluate expression (⏎) or add a ..

```
{
    1 usage
    private static int add1(int a){   a: 1
        int b = a+1;   a: 1
        int c = b+1;
        return b;
    }

    public static void main( String[] args )
```

```
"main"@1 in group "main": RUNNING
add1:10, App (org.example)
main:18, App (org.example)
```

```
a = 1
```

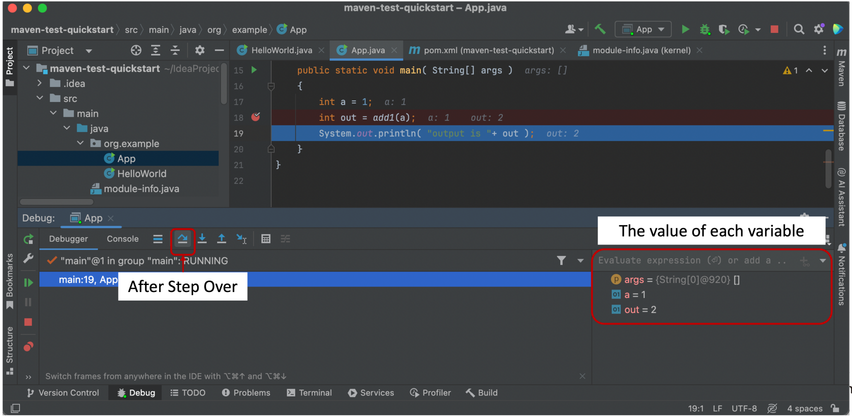
Switch frames from anywhere in the IDE with ⌥⌘↑ and ⌥⌘↓

After Step Into

The value of each variable

Debugging with IntelliJ IDEA.

- **Step over**. The Step over action does not enter into function instead, it will jump to the next line of code.


maven-test-quickstart – App.java
maven-test-quickstart 〉 src 〉 main 〉 java 〉 org 〉 example 〉 App
Project
maven-test-quickstart ~/IdeaProjec
.idea
src
main
java
org.example
App
HelloWorld
module-info.java
HelloWorld.java
App.java
pom.xml (maven-test-quickstart)
module-info.java (kernel)
Debug:
App
Debugger
Console
Version Control
Debug
TODO
Problems
Terminal
Services
Profiler
Build
19:1
LF
UTF-8
4 spaces
Maven
Database
AI Assistant
Notifications
Bookmarks
Structure

```
public static void main( String[] args )   args: []
{
    int a = 1;   a: 1
    int out = add1(a);   a: 1    out: 2
    System.out.println( "output is "+ out );   out: 2
}
}
```

"main"@1 in group "main": RUNNING

main:19, App

After Step Over

The value of each variable

Evaluate expression (⏎) or add a ..

args = {String[0]@920} []

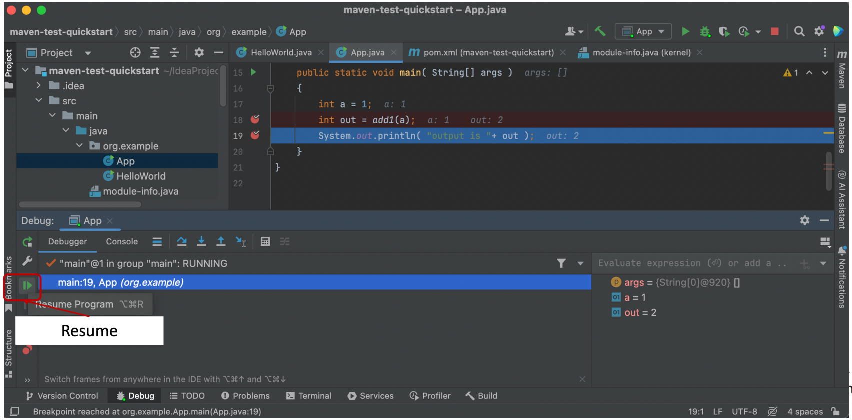
a = 1

out = 2

Switch frames from anywhere in the IDE with ⌥⌘↑ and ⌥⌘↓

Debugging with IntelliJ IDEA.

- *Resume Program*. The Resume Program action will continue the execution of the program by ignoring the breakpoint.


maven-test-quickstart – App.java
maven-test-quickstart 〉 src 〉 main 〉 java 〉 org 〉 example 〉 App
Project
maven-test-quickstart ~/IdeaProjec
.idea
src
main
java
org.example
App
HelloWorld
module-info.java
HelloWorld.java
App.java
pom.xml (maven-test-quickstart)
module-info.java (kernel)
Maven
Database
AI Assistant
Notifications
Bookmarks
Structure
Debug:
App
Debugger
Console
Evaluate expression (⏎) or add a ..
Resume Program ⌥⌘R
Resume
Version Control
Debug
TODO
Problems
Terminal
Services
Profiler
Build
19:1 LF UTF-8 4 spaces

```
public static void main( String[] args )  args: []
{
    int a = 1;  a: 1
    int out = add1(a);  a: 1   out: 2
    System.out.println( "output is "+ out );  out: 2
}
}
```

"main"@1 in group "main": RUNNING

main:19, App *(org.example)*

args = {String[0]@920} []
a = 1
out = 2

Switch frames from anywhere in the IDE with ⌥⌘↑ and ⌥⌘↓

Breakpoint reached at org.example.App.main(App.java:19)

Java Project Management Tools

Maven is a powerful project management tool that is based on POM (project object model). It is used for project build, dependency, and documentation.

The main functions include:

- *Adding a set of Jars in each project.*
- *Creating the right project structure.*
- *Building and Deploying the project.*

Java Project Management Tools: Maven

Create a Maven Project with IntelliJ IDEA

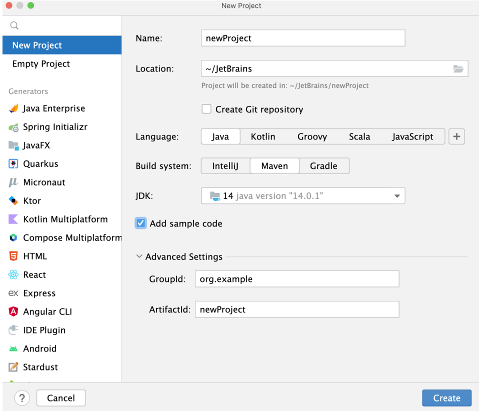
1. *Make sure you have downloaded the* **Ultimate** *version of IntelliJ IDEA (see above)*.

   IntelliJ IDEA supports a fully-functional integration with Maven that helps you automate your building process. You can easily create a new Maven project, open and sync an existing one, add a Maven support to any existing IntelliJ IDEA project, configure and manage a multi-module project.

Java Project Management Tools: Maven

Create a Maven Project with IntelliJ IDEA

1. *Make sure you have downloaded the* **Ultimate** *version of IntelliJ IDEA (see above).*
2. *Create a new Maven project (check this page for the detailed step).*



New Project
New Project
Empty Project
Generators
Java Enterprise
Spring Initializr
JavaFX
Quarkus
Micronaut
Ktor
Kotlin Multiplatform
Compose Multiplatform
HTML
React
Express
Angular CLI
IDE Plugin
Android
Stardust
?
Cancel
Create

Name: newProject

Location: ~/JetBrains

Project will be created in: ~/JetBrains/newProject

☐ Create Git repository

Language: Java | Kotlin | Groovy | Scala | JavaScript | +

Build system: IntelliJ | Maven | Gradle

JDK: 14 java version "14.0.1"

☑ Add sample code

Advanced Settings

GroupId: org.example

ArtifactId: newProject

Java Project Management Tools: Maven

Create a Maven Project with IntelliJ IDEA

1. *Make sure you have downloaded the* **Ultimate** *version of IntelliJ IDEA (see above).*
2. *Create a new Maven project (check this page for the detailed step).*
3. *(optional) Change the Maven Repository.*

   A maven repository is **a directory of packaged JAR file** with pom.xml file.

   Maven searches for dependencies in the repositories (check this page for the detailed step).

Maven *pom.xml* file.

POM is an acronym for *Project Object Model*. The pom.xml file contains information of project and configuration information for the maven to build the project.

1. Elements of maven pom.xml file.

Element	Description
project	It is the root element of pom.xml file.
modelVersion	It specifies the modelVersion. It should be set to 4.0.0.
groupId	It specifies the id for the project group.
artifactId	It specifies the id for the artifact (project). An artifact is something that is either produced or used by a project.
version	It is the sub element of project. It specifies the version of the artifact under given group.

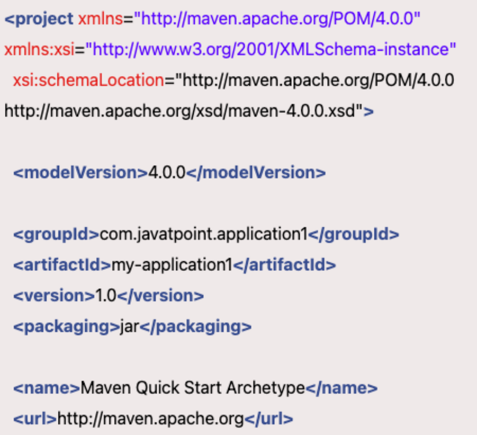
Maven *pom.xml* file.

POM is an acronym for *Project Object Model*. The pom.xml file contains information of project and configuration information for the maven to build the project.

1. Elements of maven pom.xml file.

Element	Description
packaging	defines packaging type such as jar, war etc.
name	defines name of the maven project.
url	defines url of the project.
dependencies	defines dependencies for this project.
dependency	defines a dependency. It is used inside dependencies.
scope	defines scope for this maven project. It can be compile, provided, runtime, test and system.

An Example of *pom.xml* file.

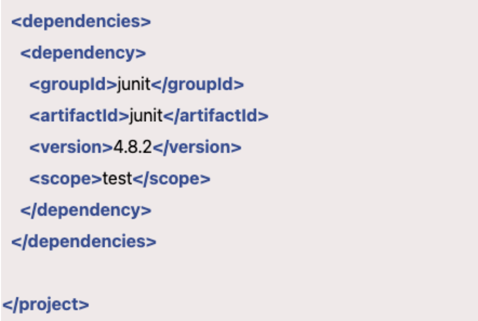


```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

 <modelVersion>4.0.0</modelVersion>

 <groupId>com.javatpoint.application1</groupId>
 <artifactId>my-application1</artifactId>
 <version>1.0</version>
 <packaging>jar</packaging>

 <name>Maven Quick Start Archetype</name>
 <url>http://maven.apache.org</url>
```



```xml
 <dependencies>
  <dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>4.8.2</version>
   <scope>test</scope>
  </dependency>
 </dependencies>

</project>
```

Question and Answering (Q&A)