

Guide for Including External Libraries in Java Projects

This guide provides step-by-step instructions for including external libraries in Java projects using BlueJ, Visual Studio Code, and IntelliJ IDEA.

BlueJ

1. **Open BlueJ:**
 - Start BlueJ.
2. **Access Preferences:**
 - Navigate to **Tools -> Preferences**.
3. **Add Libraries:**
 - Go to the **Libraries** tab.
 - Click **Add** and navigate to the **lib** directory where your JAR files are located.
 - Each library in the provided **lib** directory has its own internal **lib** directory. Select all the JAR files inside each internal **lib** directory.
4. **Reset Java Virtual Machine:**
 - Either close and reopen BlueJ or go to **Tools -> Reset Java Virtual Machine** to apply the changes.

Note: If you are using Mac or Linux, you may need to download the appropriate version of JavaFX for your machine. You can download JavaFX from Gluon [here](#).

NetBeans

1. **Open NetBeans:**
 - Start NetBeans.
2. **Create or Open a Java Project:**
 - Open your existing Java project or create a new one.
3. **Add Libraries:**
 - Right-click on the project in the **Projects** tab.
 - Select **Properties**.
 - Go to **Libraries** -> **Compile**.
 - Click **Add JAR/Folder** and navigate to the **lib** directory where your JAR files are located.
 - Each library in the provided **lib** directory has its own internal **lib** directory. Select all the JAR files inside each internal **lib** directory and add them.
4. **Apply and Close:**
 - Click **OK** to close the Project Properties dialog.

Note: Similar to the other environments, if you are using Mac or Linux, you may need to download the appropriate version of JavaFX for your machine. You can download JavaFX from Gluon [here](#).

Visual Studio Code

1. **Open Visual Studio Code:**
 - Start Visual Studio Code.
2. **Install Extensions:**
 - Ensure you have the Java Extension Pack installed. If not, go to the Extensions view (**Ctrl+Shift+X**) and search for **Java Extension Pack**, then install it.
3. **Create or Open a Java Project:**
 - Open your Java project folder or create a new Java project.
4. **Configure Classpath:**
 - Create a folder named **lib** in your project directory (if it doesn't already exist).
 - Place all the necessary JAR files into the **lib** folder.
5. **Update .classpath File:**
 - Open the **.classpath** file in the root directory of your project.
 - Add entries for each JAR file. For example:

```
<classpath>
  <classpathentry kind="src" path="src"/>
  <classpathentry kind="lib" path="lib/library1.jar"/>
  <classpathentry kind="lib" path="lib/library2.jar"/>
  <!-- Add entries for all JAR files -->
</classpath>
```

6. Refresh Project:

- Right-click on your project in the Explorer view and select **Refresh** to ensure the changes are recognized.

IntelliJ IDEA

1. Open IntelliJ IDEA:

- Start IntelliJ IDEA.

2. Open Project:

- Open your existing Java project or create a new one.

3. Access Project Structure:

- Navigate to **Settings** -> **Project Structure**.

4. Add Libraries:

- In the Project Structure dialog, go to **Libraries**.
- Click the **+** (Add) button, then select **Java**.
- Navigate to the **lib** directory and add all the JAR files.

5. Apply and Close:

- Click **Apply** and then **OK** to close the Project Structure dialog.

Note: Similar to BlueJ, if you are using Mac or Linux, you may need to download the appropriate version of JavaFX for your machine. You can download JavaFX from Gluon [here](#).

Maven

1. **Open Project in NetBeans or IntelliJ IDEA:**
 - Open your Maven project in either NetBeans or IntelliJ IDEA.
2. **Verify `pom.xml`:**
 - Ensure that the provided `pom.xml` file is present in the project root directory.
3. **Build the Project:**
 - In NetBeans:
 - Right-click on the project in the `Projects` tab.
 - Select `Build` to build the project.
 - In IntelliJ IDEA:
 - Click on the `Maven` tool window on the right side.
 - Click the `Reimport All Maven Projects` icon or right-click on your project and select `Maven -> Reimport`.

Note: The required `pom.xml` file is included with the assignment files. It contains the necessary dependencies for Apache Commons Math, Apache Commons Numbers, and JavaFX.