

Install Eclipse for Java and C++ Development

Ronald Mak

Department of Computer Engineering

Department of Computer Science

Department of Applied Data Science

January 13, 2020

Revised: February 16, 2020

Introduction

Eclipse is a popular integrated development environment (IDE) for developing Java and C++ applications. You can install plugins that provide special features for specialized development frameworks, such as ANTLR 4.

You can download, configure, build, and install Eclipse on Ubuntu, MacOS X, or Windows 10. If you want to run on Ubuntu, first read the following tutorials:

- “Install Ubuntu on Windows 10 and VirtualBox”
<http://www.cs.sjsu.edu/~mak/tutorials/InstallUbuntu.pdf>
- “Configure Ubuntu for Software Development”
<http://www.cs.sjsu.edu/~mak/tutorials/ConfigureUbuntu.pdf>

The process to install Eclipse on any of the three platforms is similar. The Ubuntu and MacOS X platforms are recommended over Windows 10, as the latter may have compatibility problems when you development programs.

Install the Java Development Kit (JDK)

Eclipse requires the Java Development Kit (JDK) which includes the **java** application for running Java programs and the **javac** compiler for compiling java programs. Go to <https://www.oracle.com/java/technologies/javase-jdk13-downloads.html> to download the installer for the JDK for your platform. This will also install **java**, **javac**, **javap**, and **jar** programs.

For MacOS X, download the latest **.dmg** image (currently **jdk-13.0.2_osx-x64_bin.dmg**) and install from that image.

For Ubuntu, download the latest Linux **.tar.gz** compressed installer file (currently **jdk-13.0.2_linux-x64_bin.tar.gz**). First unzip the file with the **gunzip** utility program, and then unarchive with the **tar** utility. For example:

```
gunzip jdk-13.0.2_linux-x64_bin.tar.gz
tar -xvf jdk-13.0.2_linux-x64_bin.tar
```

which will create a **jdk-13.0.2** directory containing Java and its utility programs.

Edit the `.bashrc` file in your home directory (it might instead be `.bash_profile` in MacOS X) to set and export `JAVA_HOME` to point to the installed `jdk` directory. For example:

```
export JAVA_HOME=/home/rmak/jdk-13.0.2
```

Otherwise, you need to type the above as a command each time you open a terminal window.

If you are using the Ubuntu subsystem of Windows 10, use a browser in Windows (such as Microsoft Edge) to download the `.tar.gz` compressed installer file. Then in the Ubuntu terminal window, use the `mv` command to move the installer file out of the Windows downloads directory into your Ubuntu home directory. Use `/mnt/c` or `/mnt/d` to access directories and files in the Windows file system. For example, from your Ubuntu home directory, enter a command like:

```
mv /mnt/c/Users/rmak/Downloads/jdk-13.0.2_linux-x64_bin.tar.gz .
```

The final dot represents the current directory (your home directory). Then proceed as described above with `gunzip` and `tar` to create the subdirectory `jdk-13.0.2` in your home directory.

Download and run the Eclipse installer

Download the Eclipse installer from <https://www.eclipse.org/downloads/packages/installer>. When you run the installer, click on *Eclipse IDE for Java Developers* to start the installation process (Figure 1).

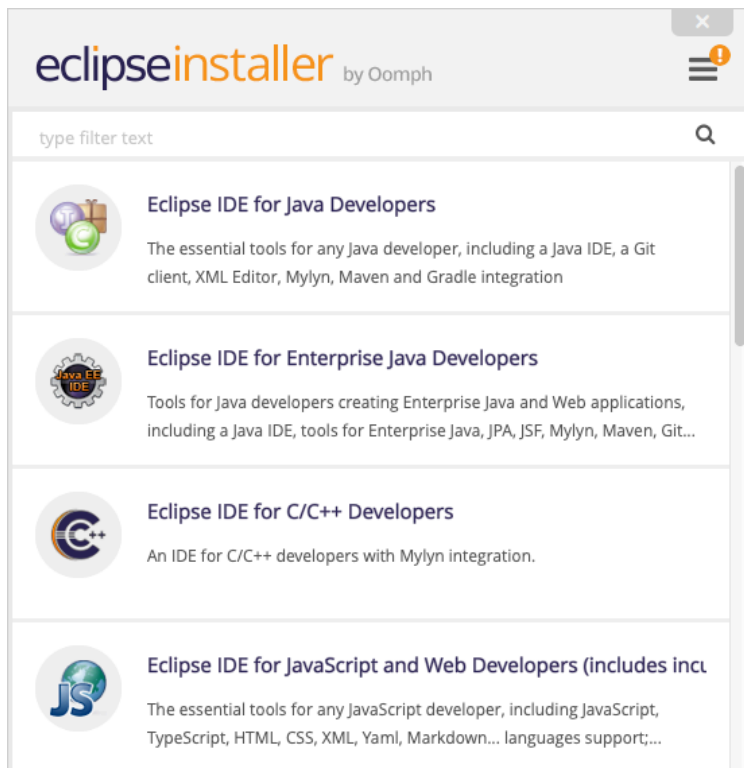


Figure 1. The Eclipse installer.

See the boxed note above about downloading installer files if you're using Ubuntu on Windows 10.

R. Mak, Install Eclipse for Java and C++ Development

When the installation is complete, launch Eclipse (Figure 2). Eclipse is now ready for you to develop Java projects.

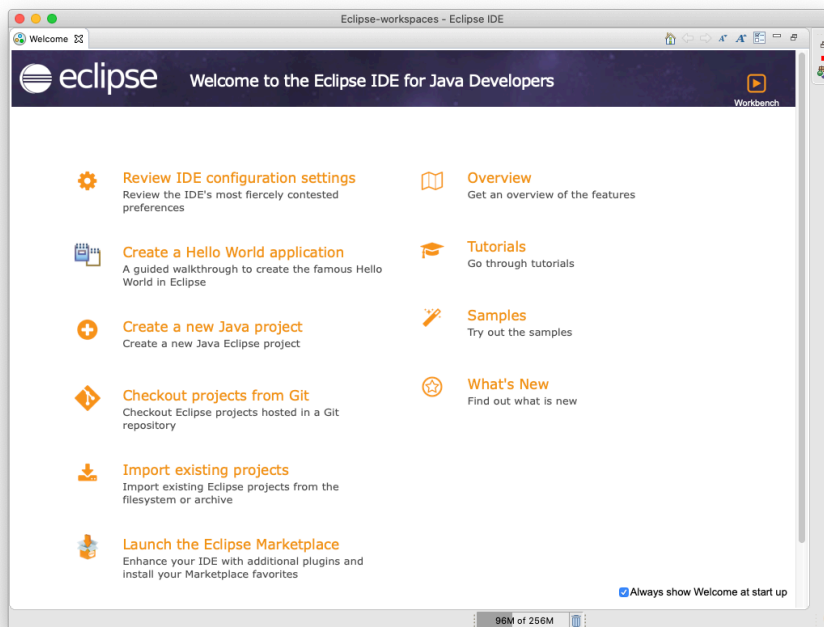


Figure 2. The initial launch of Eclipse.

Create a Java project

If you intend to develop Java programs, you're all set. If it's still open, close the *Welcome* tab.

Right-click in the **Project Explorer** panel at the left. From the context menu, choose *New | Java Project*.

Enter a project name, such as "MergeSort" (Figure 3). Click the *Finish* button.

R. Mak, Install Eclipse for Java and C++ Development

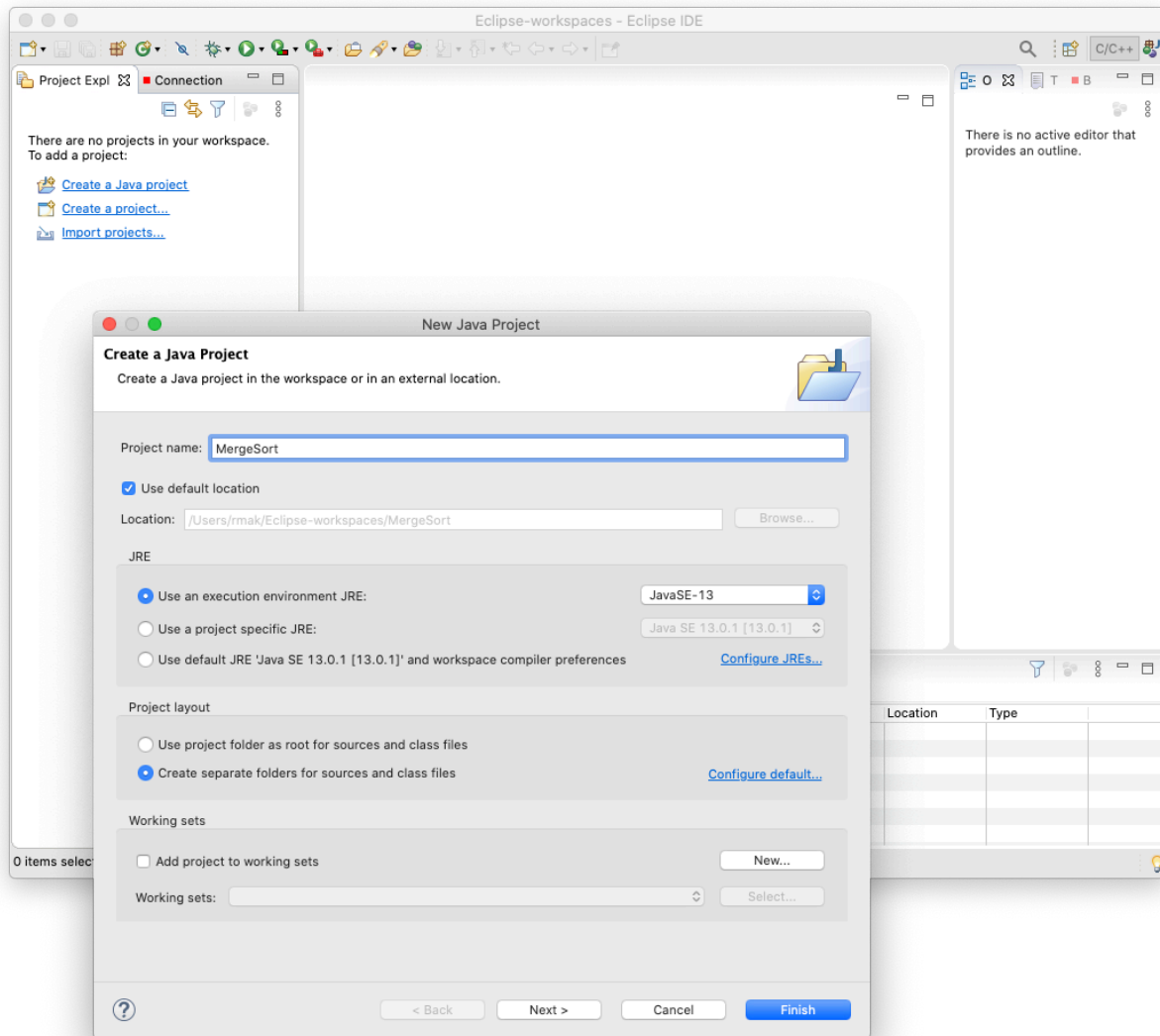
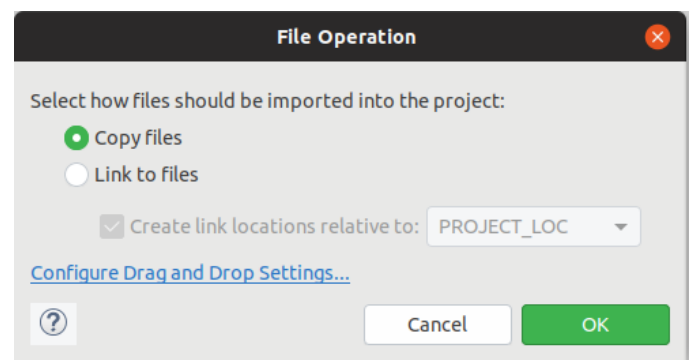


Figure 3. Create your Java project.

Right-click the project name in the **Project Explorer** panel and select *New* from the context menu to create a class or a package. You can also add files to the project by dragging the files' names from an external file browser onto *src* under the project name. Select the *Copy files* option and click the *OK* button (Figure 4).

Figure 4. Copy files into a project.



Now you're ready to edit, compile, and run your Java program (Figure 5).

R. Mak, Install Eclipse for Java and C++ Development

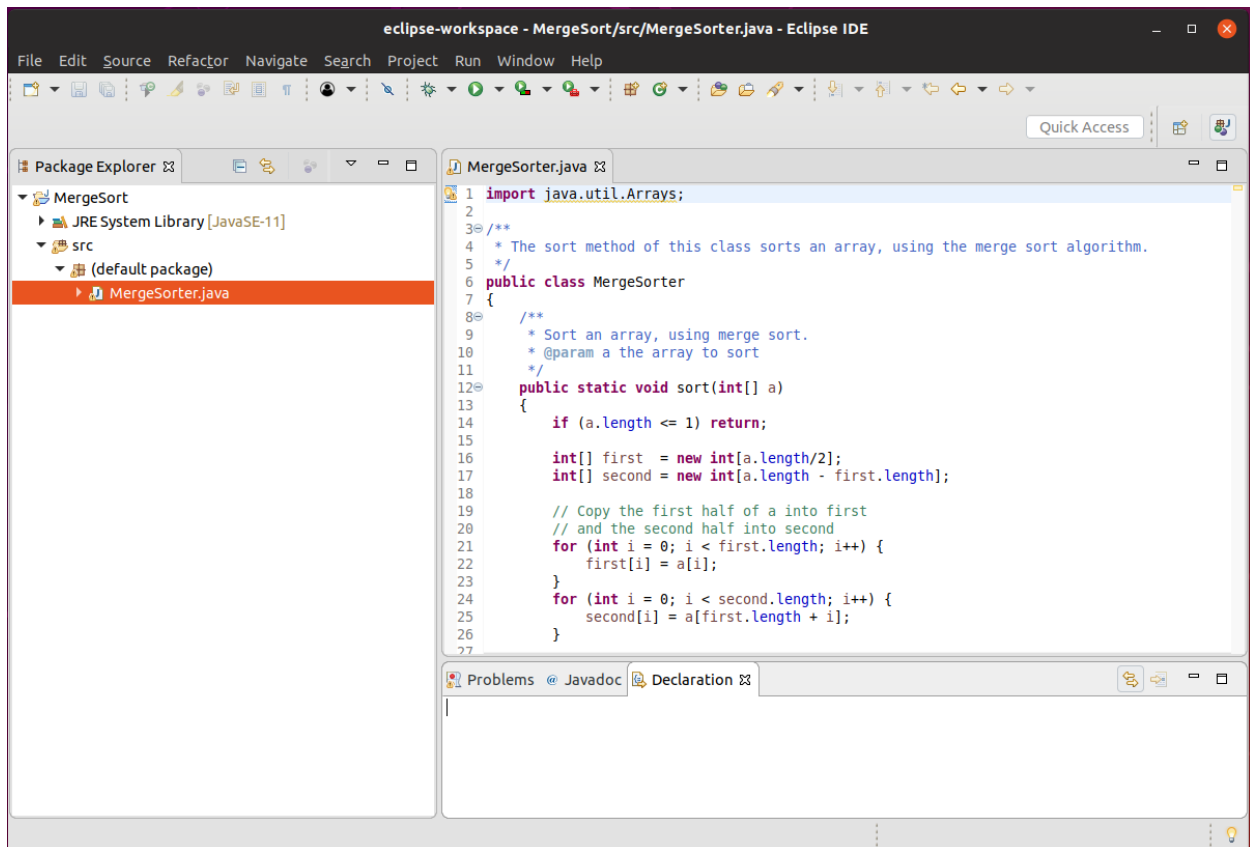


Figure 5. Editing a Java program.

If you are going to develop only Java projects, then you are done installing Eclipse, and you do not need to read the rest of this tutorial.

Install the C++ Development Toolkit (CDT)

Install the CDT plugin if you intend to use Eclipse to develop C++ projects.

From the *Help* dropdown menu at the top of the Eclipse window, choose *Install New Software ...* A dialog box opens for you to choose which plugins to install (Figure 7). Enter the URL <http://download.eclipse.org/tools/cdt/releases/9.10> into the *Work with* box and press the return key. Select *CDT Main Features* (Figure 6).

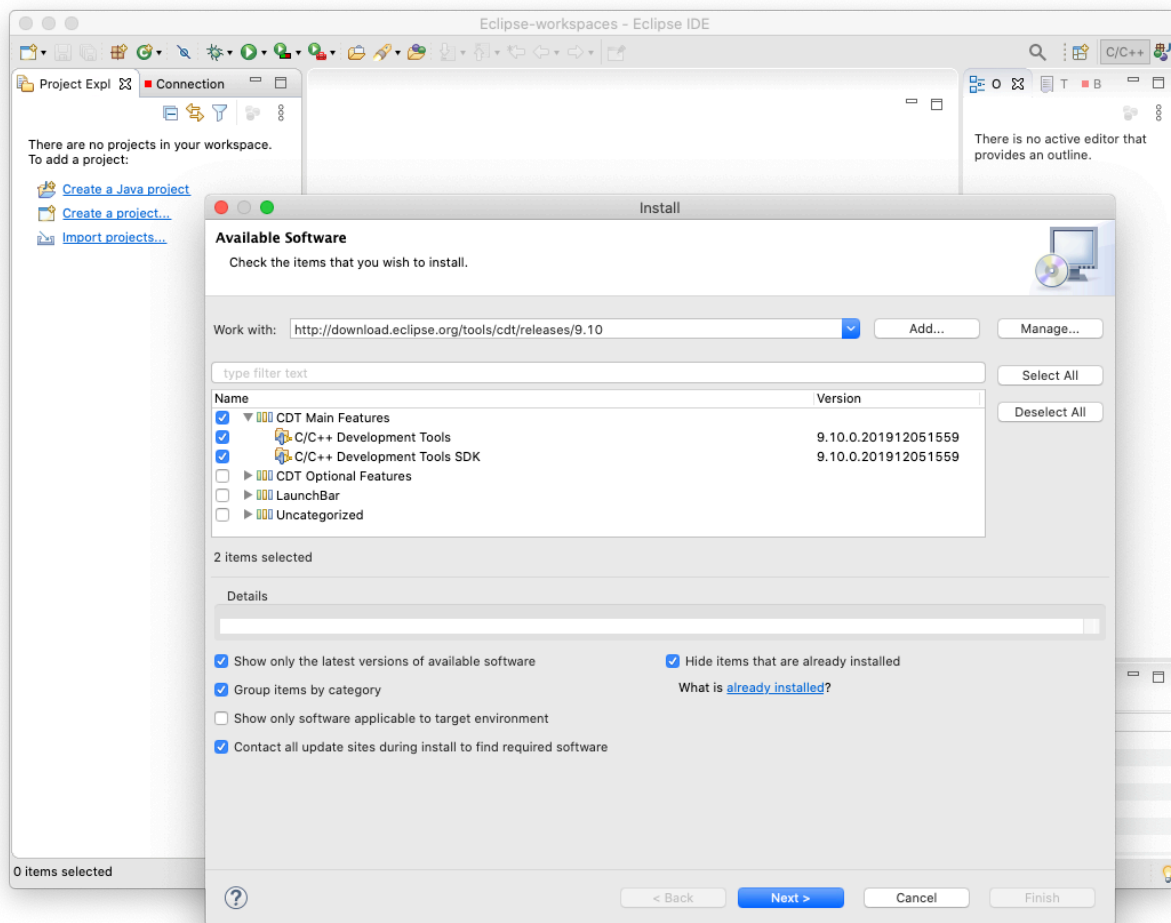


Figure 6. The dialog box for installing plugins.

Click the *Next* button twice, accept the terms of the license, and click the *Finish* button. Wait for the plugin to finish installing. Eclipse will then prompt you to restart it.

Create a C++ project

After Eclipse restarts, close the *Welcome* tab if it's still open.

From the *Window* dropdown menu, choose *Perspective | Open Perspective | Other ... | C/C++* (Figure 7). Click the *Open* button.

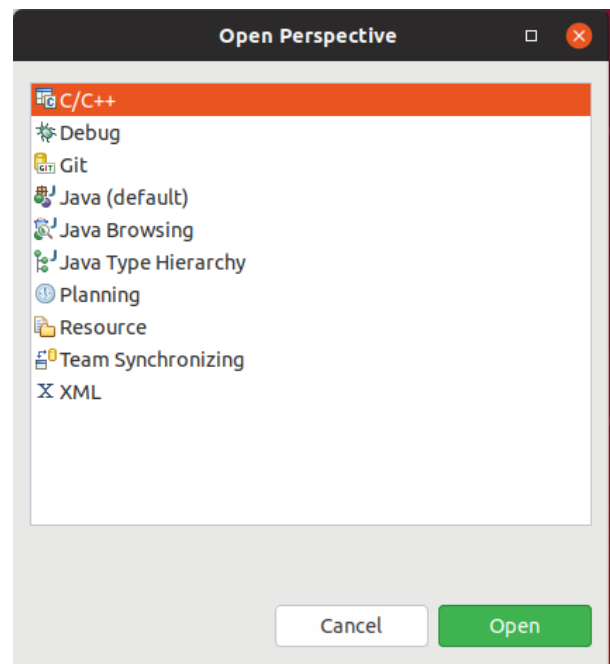


Figure 7. Open the C/C++ perspective.

Right-click in the *Project Explorer* panel at the left. From the context menu, choose *New | C/C++ Project*. Choose *C++ Managed Build* and click the *Next* button (Figure 8).

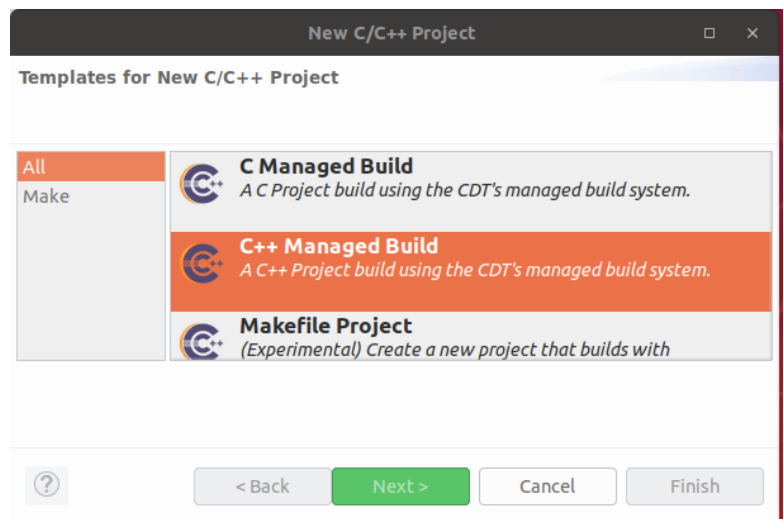


Figure 8. Choose the C++ managed build.

Enter a name for your project. *Select Empty Project* and *Linux GCC*. Do not select a “cross compiler” in the “Toolchains” box. Click the *Finish* button to create your C++ project (Figure 9).

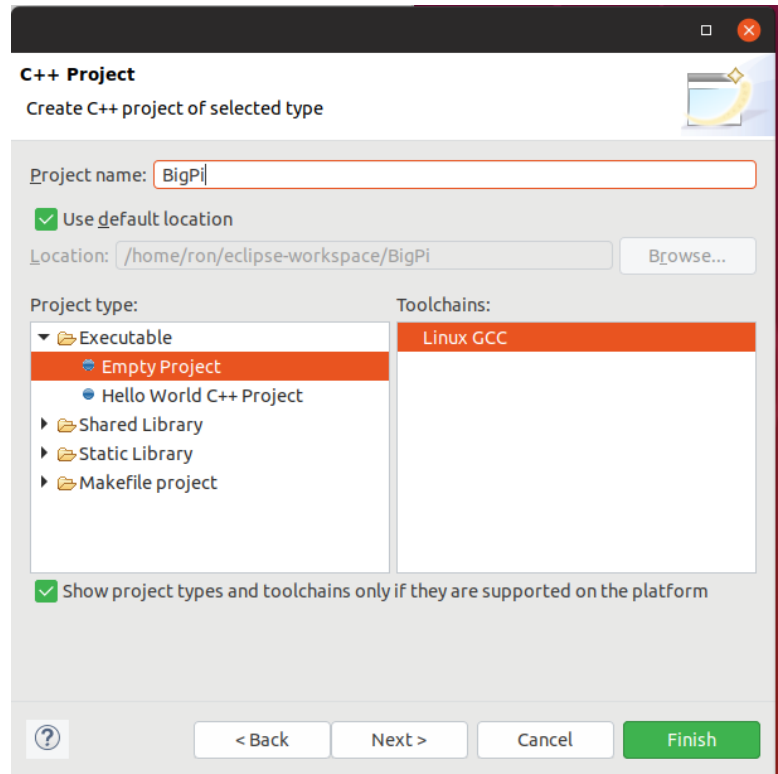
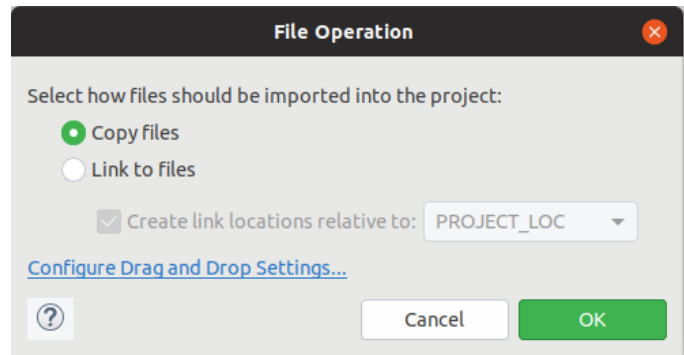


Figure 9. Create your C++ project.

Right-click the project name in the **Project Explorer** panel and select *New* from the context menu to create a class, a header file, or a C++ source file. You can also add files to the project by dragging the files' names from an external file browser onto the project name. Select the *Copy files* option and click the *OK* button (Figure 10).

Figure 10. Copy files into a project.



Now you're ready to edit, compile, and run your C++ program (Figure 11).

R. Mak, Install Eclipse for Java and C++ Development

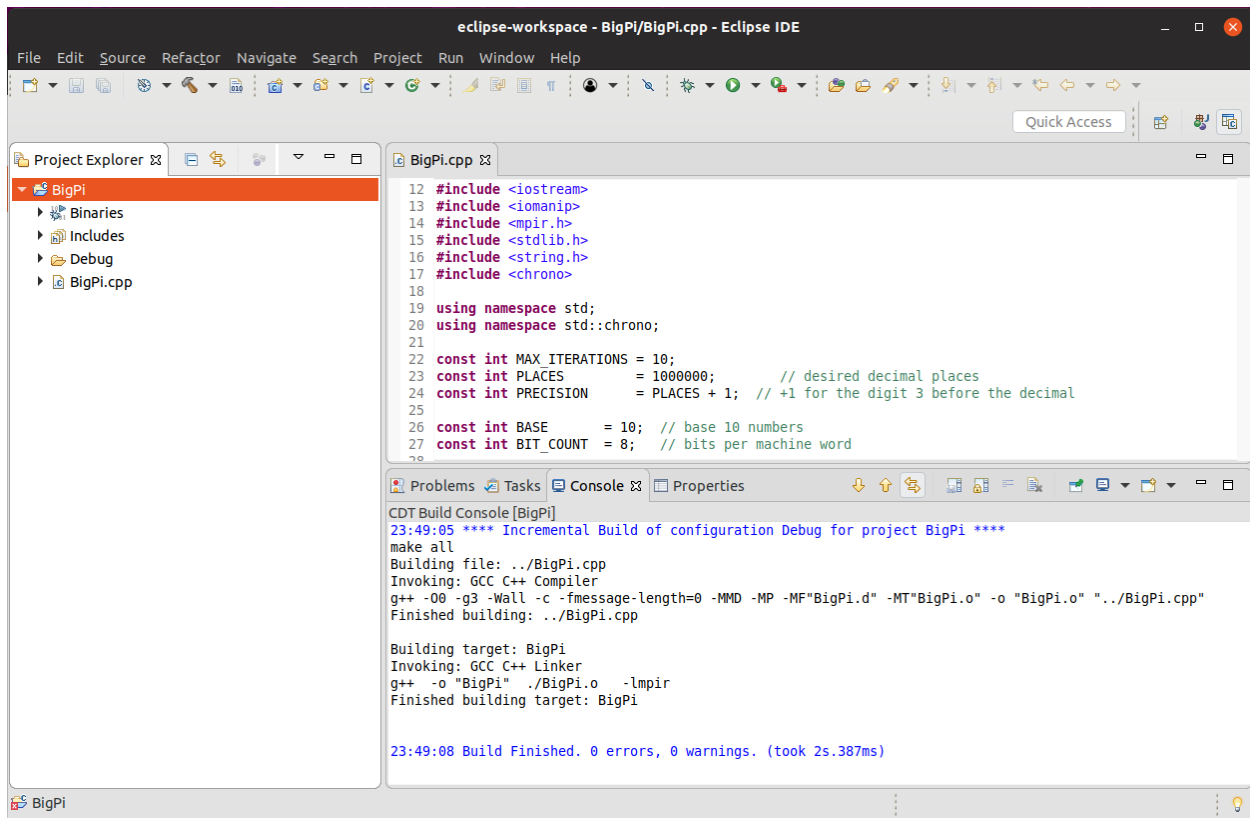


Figure 11. Editing and compiling a C++ program.