

Using the BlueJ IDE

The authors of the *Advanced Placement Computer Science: Java Programming* course recommend that students attempt some or all of the projects that are provided as part of the curriculum in order to gain experience both in the core areas covered by the online course and in other relevant (though not required) subsidiary areas.

These projects are best experienced in an Integrated Development Environment (IDE) that includes a code editor and debugger. One such IDE is the *BlueJ* IDE, published free by **BlueJ.org**. This document includes instructions for downloading and using the *BlueJ* IDE.

How to Install the BlueJ IDE

If you have already installed the *BlueJ* IDE, please proceed directly to the *How to Use the BlueJ IDE* section on the next page.

The *BlueJ* Integrated Development Environment is published by **BlueJ.org**. Versions of this application suitable for various operating systems, including Windows, Linux, and MacOS X, are available free on the internet.

Installation is a two-stage process. First, you must install the Java 2 Standard Edition Development Kit (JDK), and then the *BlueJ* IDE. Starting with Mac OS X 10.1.1, the operating system includes a suitable Java installation, so Macintosh users running Mac OS X may proceed directly to the *Step 2: Install BlueJ* section below. (It is worthwhile, however, using the Macintosh Software Update feature to check whether a newer Java installation has become available.)

Step 1: Install Java

If you are not using a Macintosh, browse to <http://java.sun.com/> and on the right-hand side of the page locate the "Popular Downloads" box. Click the **Java SE** link. In the *Downloads | Latest Release* list that appears, click the **Download** button for the most recent non-beta Java SE Development Kit (JDK) release. (At the time of writing, this is **JDK 6 Update 14**.) Choose the by-itself version, not one that comes with other things such as JavaFX SDK or Java EE. On the download page, select your operating system from the *Platform* drop-down box, leave the *Language* selection as "Multi-language", and check the box to indicate that you accept the appropriate License Agreement. Then click the **Continue** button. Check the box next to the file description in the *Available Files* list, and finally, initiate the download of the installation setup program appropriate for your operating system by clicking on the file name. To complete the installation, run or execute the downloaded installation file using the appropriate technique for your operating system.

Step 2: Install BlueJ

Browse to <http://www.bluej.org/> and click the "download" link in the navigation bar on the left of the page. To initiate downloading of the installation package, click the filename next to the mention of your operating system under the "Download the

installation package” heading. Once the download is complete, click the “go to the Installation Instructions” link near the bottom of the *BlueJ* web site’s download page, and follow the instructions that are displayed for your operating system.

How to Use the BlueJ IDE

Hello World!

When learning a new programming language or investigating a new development environment for the first time, it is sensible to begin by writing a program that is trivial. It is traditional in computer science to begin with a “Hello World!” program whose only task is to print the words “Hello World!” somewhere on the screen.

We will write the “Hello World!” program using *BlueJ*. Begin by starting the *BlueJ* IDE. The basic application workspace is shown in Figure 1 below.

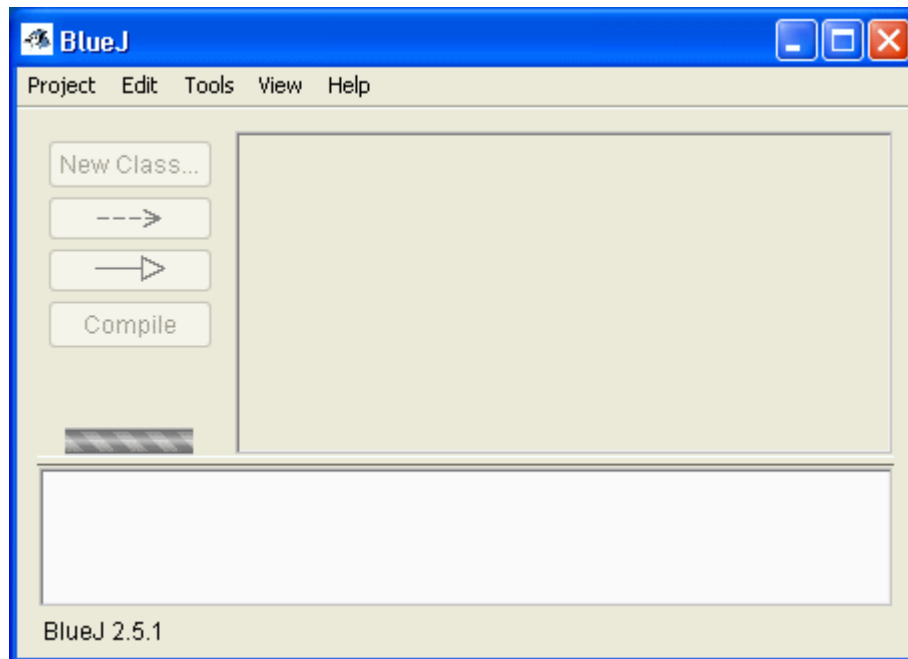


Figure 1: The BlueJ Application Workspace.

From the **Project** menu, choose **New Project....** A *New Project* dialog appears. This is very similar to a standard *Save As* dialog. Navigate through your computer’s file system in the usual way until the *Look In* selector at the top of the dialog displays the name of the folder in which you want to save your *BlueJ* projects. (If you wish, this can be a new folder created using the “Create New Folder” button.) Enter the name “Hello World” (without the quote marks) into the *File name* edit box and click the **Create** button. The project name is added to the title of the *BlueJ* application window.

Click the **New Class...** button in the *BlueJ* application workspace. A *Create New Class* dialog appears (see Figure 2).

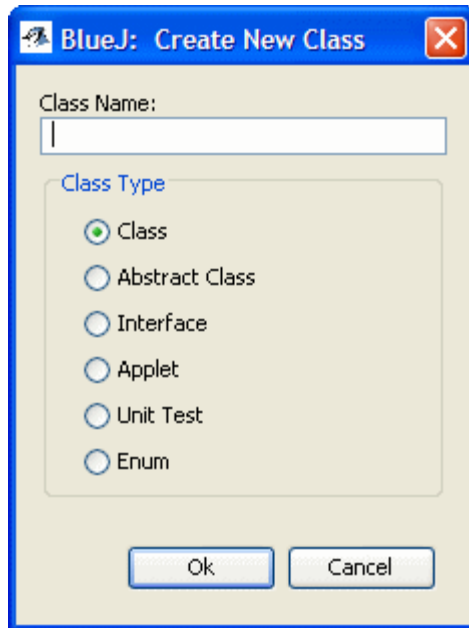


Figure 2: Creating a New Class.

Enter the name "Main" (without the quote marks) into the *Class Name* edit box. Leave the *Class Type* selection as the default "Class" and click the **Ok** button. A colored rectangular icon representing the newly-created `Main` class appears in the main *BlueJ* window (see Figure 3).

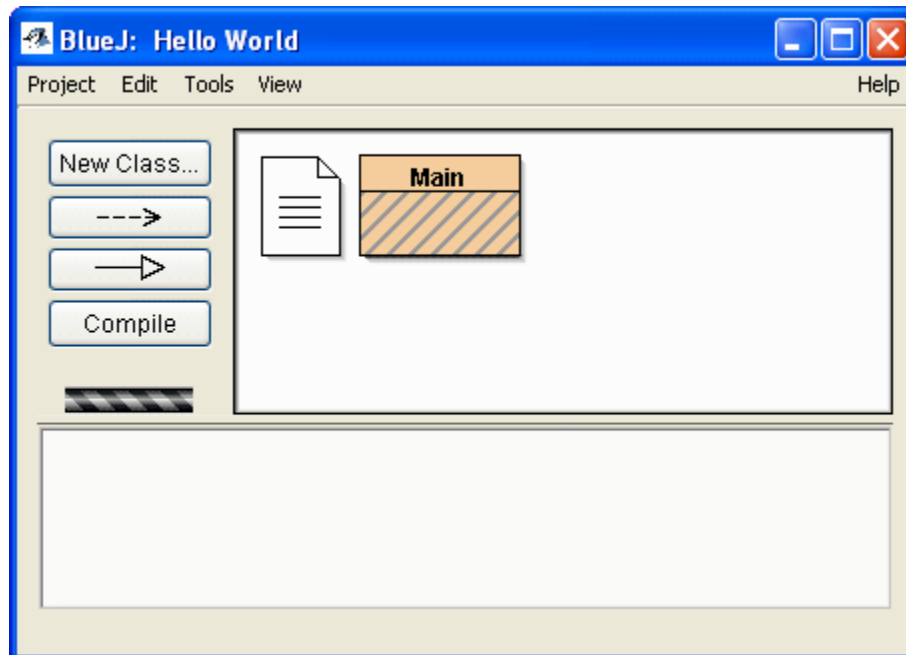


Figure 3: The Main Class Icon.

Right-click on the `Main` class icon.¹ On the resulting context menu, choose the **Open Editor** item. An editor window containing sample code for the `Main` class appears (see Figure 4).

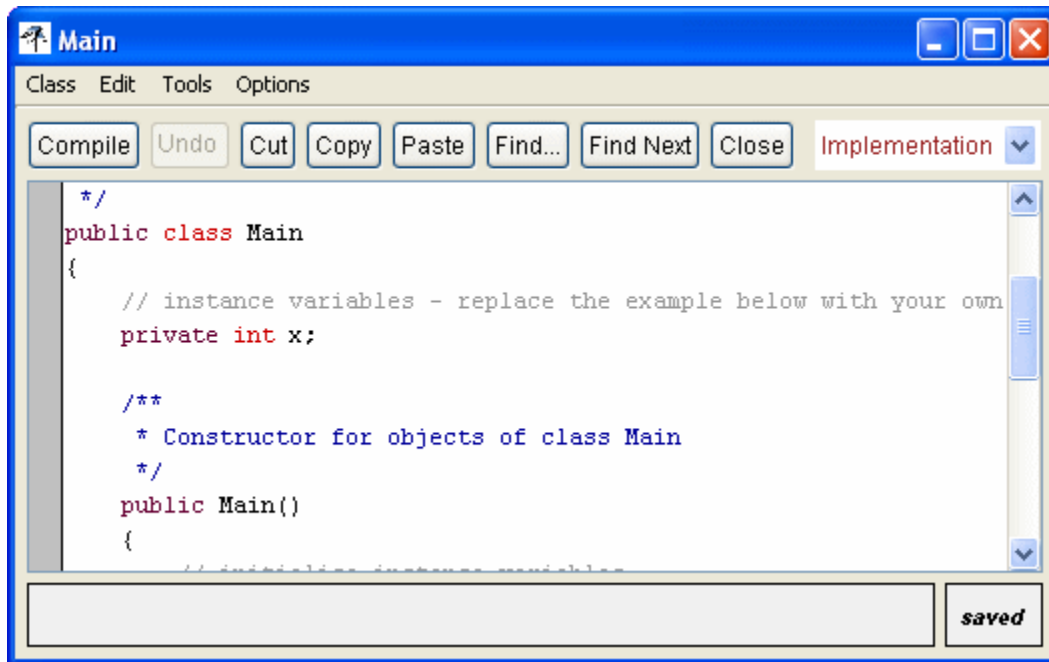


Figure 4: The Main Class Editor Window.

Edit the sample code in the editor window until it reads as shown in Figure 5 on the next page. Click the **Close** button on the editor window. The changes you have made to the code are automatically saved.

Before you can run your “Hello World!” program, you must compile the code. To do this, click the **Compile** button on the main *BlueJ* window. After a short pause, the `Main` class icon loses its striping, thereby indicating that the class has compiled successfully.

When it is run, your program will cause the phrase “Hello World!” to be printed somewhere. That “somewhere” is the *BlueJ* Terminal. To reveal the Terminal window, choose the **Show Terminal** item on the **View** menu. In order to run the program, you must execute the `main` method of the `Main` class. You do this by right-clicking on the `Main` class icon and then choosing the signature — `void main(args)` — of the `main` method on the resulting context menu. A *Method Call* dialog appears (see Figure 6 on the next page). Click the **Ok** button. The *Method Call* dialog goes away and the phrase “Hello World!” appears on the *BlueJ* Terminal.

Congratulations! You have compiled and run the “Hello World!” application.

¹ Macintosh users should read “right-click” as *ctrl-click*.

```
public class Main
{
    /**
     * Constructor for objects of class Main
     */
    public Main()
    {
    }

    /**
     * The main method
     */
    public static void main(String[] args)
    {
        System.out.println( "Hello World!" );
    }
}
```

Figure 5: Edit the Main Class Definition.

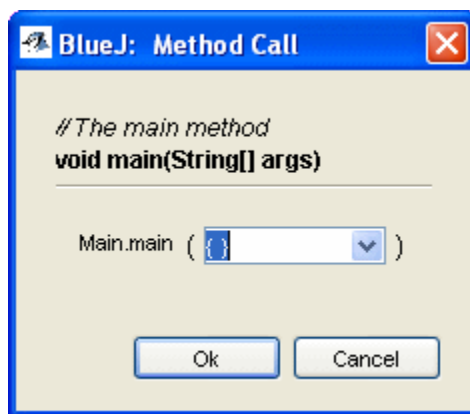


Figure 6: The Method Call Dialog.

Using the BlueJ Debugger

Reopen the `Main` class editor window by right-clicking the `Main` class icon and choosing the **Open Editor** item on the resulting context menu. Then edit the `main` method and above it add a new static method, so that the code in the body of the class definition looks like this:

```
public static void test( int n )
{
    System.out.println( n );
}
```

```
public static void main(String[] args)
{
    for ( int i = 0 ; i < 5 ; i++ )
    {
        test( i );
    }
}
```

Save and compile your changed definition, this time by clicking the **Compile** button on the editor window. Test that your program runs by right-clicking the `Main` class icon, choosing the `void main(args)` item on the context menu, and clicking the **Ok** button of the *Method Call* dialog. The integers from 0 through 4 should be displayed on the *BlueJ* Terminal, each on a separate line.

Next, place a breakpoint on the first line of the `for` loop by clicking once anywhere in that line and then choosing the **Set/Clear Breakpoint** item on the editor window's **Tools** menu. A STOP sign appears in the left margin (see Figure 7).

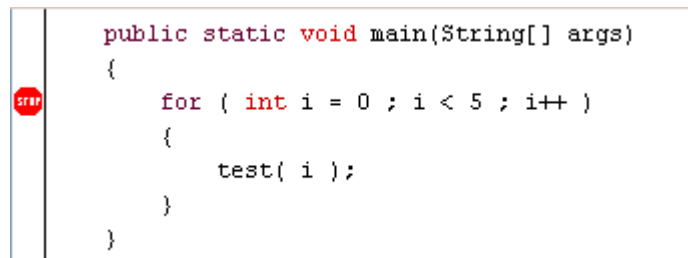


Figure 7: Setting a Breakpoint.

Open the *BlueJ* Debugger Window by choosing the **Show Debugger** item on the main *BlueJ* window's **View** menu. Run the program by right-clicking the `Main` class icon, choosing the `void main(args)` item on the context menu, and clicking the **Ok** button on the *Method Call* dialog. The Debugger window activates itself (see Figure 8 on the next page) and the first line of the `for` loop in the `Main` class editor window is highlighted in yellow to indicate that the program has executed to this point, and has stopped at this line.

To execute statements, repeatedly click the **Step** button on the Debugger window. If the statement involves a method call, and you would like to single-step into the method body, click the **Step Into** button instead. As you step through the program, watch how the information displayed in the Debugger window changes.

Getting More Help ...

If you are connected to the internet, an easy-to-read tutorial is available by choosing the **BlueJ Tutorial** item on the **Help** menu of the main *BlueJ* window. A more extensive reference manual can be downloaded from the *BlueJ* web site, which is accessible from the **BlueJ Web Site** item on the **Help** menu.

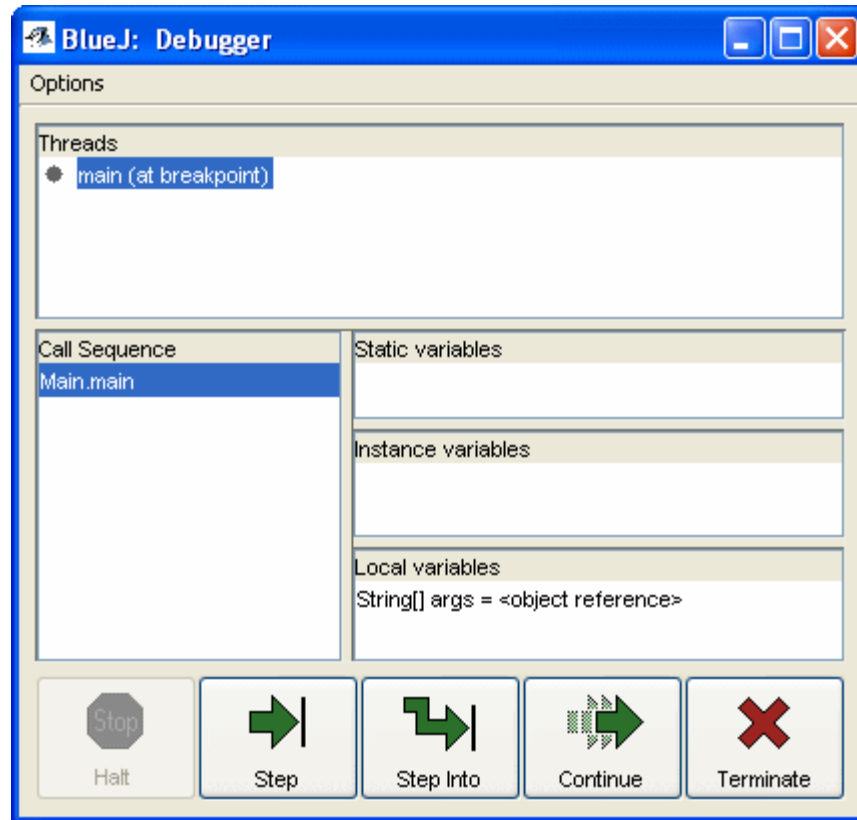


Figure 8: The *BlueJ* Debugger Window.