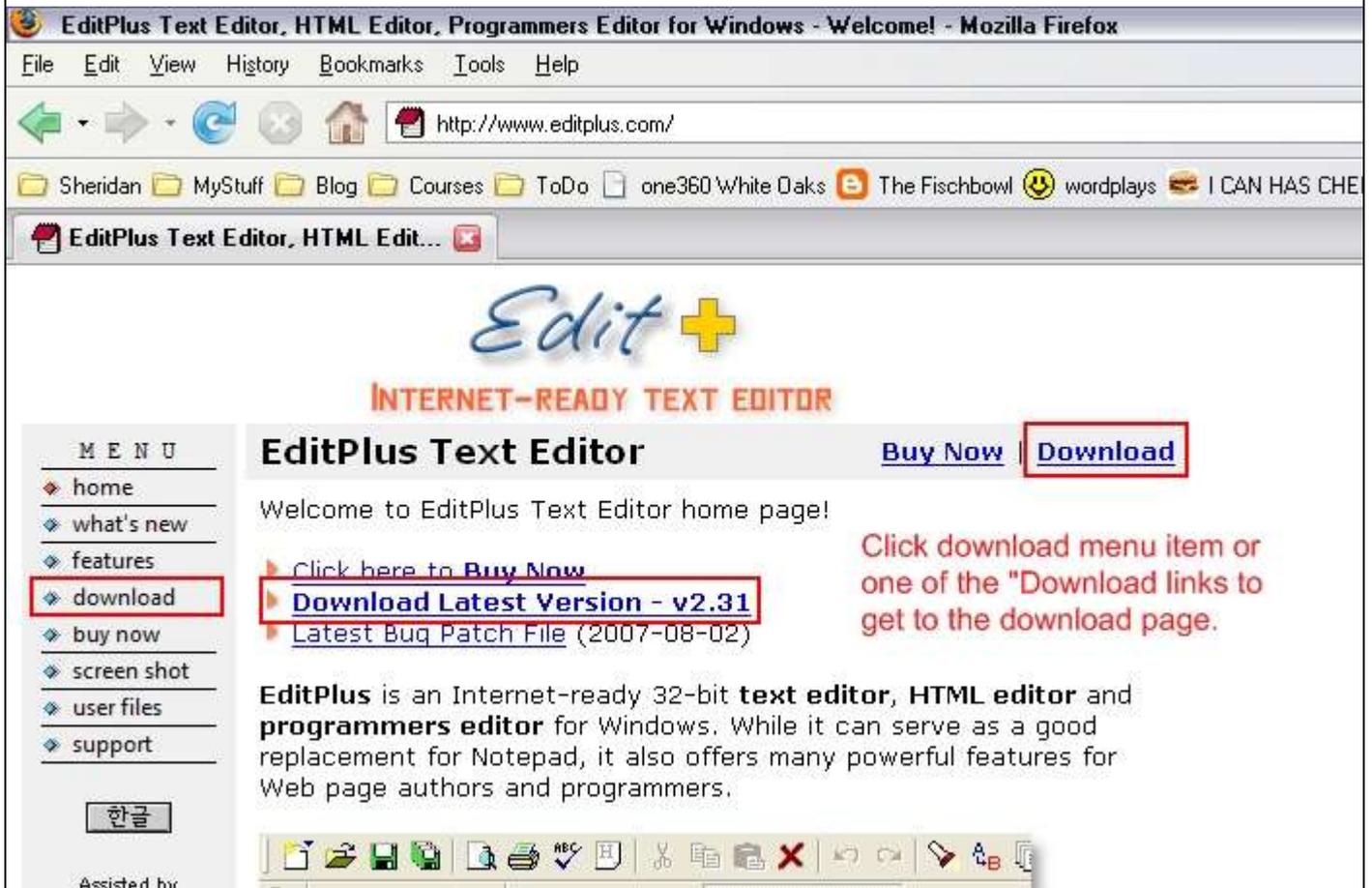


EditPlus Setup for Java

These instructions will show you how to download and set up EditPlus to compile and run Java programs. You can also make coding Java easier by changing or adding certain configuration files and templates.

Downloading and Installing

1. To download the latest version of EditPlus, go to editplus.com. You will see a variety of ways to get to the download page:



you might see a version newer than the one in the picture!

2. Once you're on the download page, click the "**Download**" link. A dialog box will appear asking if you want to save the EditPlus install file. Click the Save button. Browse to a preferred location on your hard drive and save the file. Don't forget where you save this file, as you'll need to find it when the download is finished.



INTERNET-READY TEXT EDITOR

- M E N U
- home
- what's new
- features
- download
- buy now
- screen shot
- user files
- support

한글

Assisted by



Download

[Buy Now](#)

EditPlus is distributed as **Shareware**. You can freely download and try it for 30 days. If you find it useful and decide to keep it over 30-day evaluation period, you should pay the registration fee.

EditPlus 2.31 Evaluation Version (1 MB)

[Download](#)

For HTTP download

[Latest Bug Pa](#)

Spell Checker f

[US English Sp](#)

[UK English Sp](#)

NOTE: Previous US English Spell Checker Setup SC US.exe contained a damaged dictionary file and could cause program crash. If you



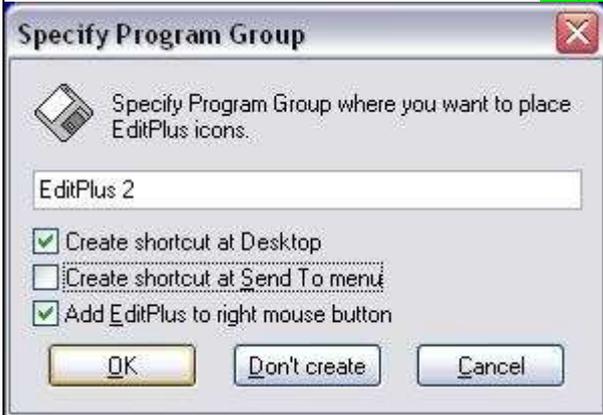
3. To install EditPlus, locate the install file and double-click it. Your default compression program should recognize the file and ask if you'd like to run the setup program. Click "Setup" or "OK" to run the setup program.

4. The next screen asks if you want to

accept the licence agreement. Click "Yes".

5. Next you will be asked to confirm the install location. If you are installing EditPlus for one of the Java courses, you can leave this as the default value. If you prefer a different location, you can select this now. Click the "Start Copy" button when you're ready to begin the installation.

6. The last screen asks you to set up the options for the Start Menu program group and the shortcut options. For your Java courses, you can leave the Program Group name alone, but you should make sure that "Create shortcut at Desktop" is selected and "Add EditPlus to right mouse button" is selected. You can unselect "Create shortcut at Send To menu.". When finished, click the OK button.



EditPlus is now installed!

Setting Up Custom Tools

There is a way to compile and run your Java source code files without using the command prompt! You can set up some menu options in EditPlus that allow you to quickly compile and run your Java programs. You might want to print these instructions so you can follow these steps. Later we will add a tool or two to this group, so make sure you keep or bookmark these instructions.

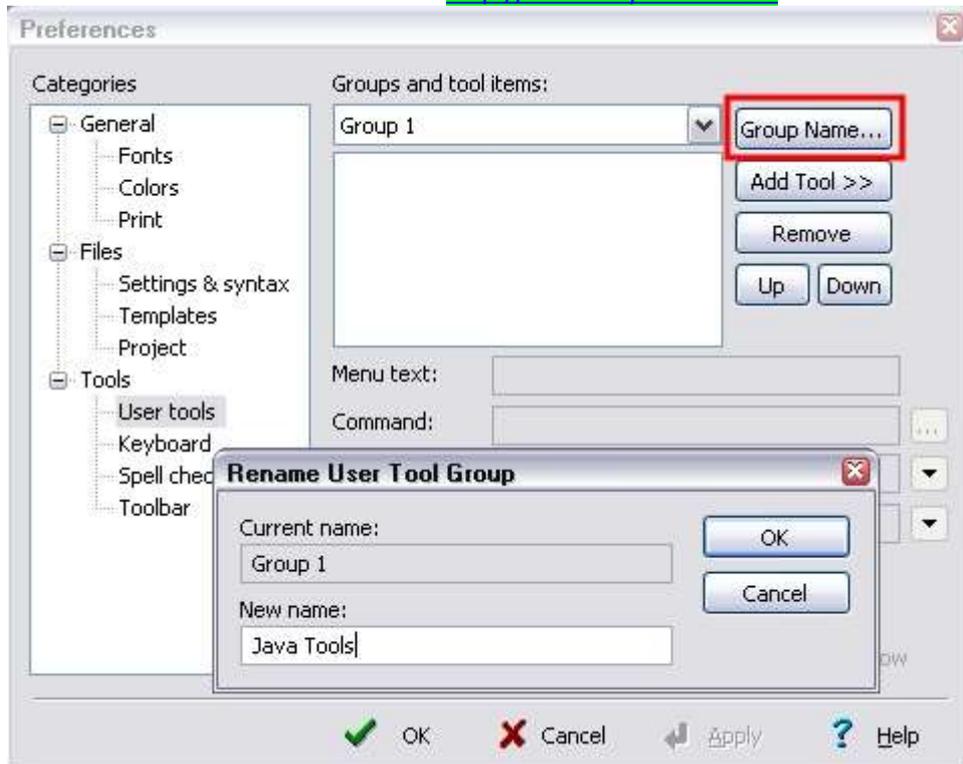
Setting up A Compile Tool

This step adds a tool to EditPlus that allows you to run the Java compile on the current file that is open. You will tell EditPlus that when you click a certain menu item or press a certain set of keys, it should run the Java compiler on the Java source code you're currently viewing. Note that EditPlus is not compiling your code; you are simply telling it where the compiler is, so that it can run the compiler for you.

Open up EditPlus. It doesn't matter if you have a new/existing document open or not.

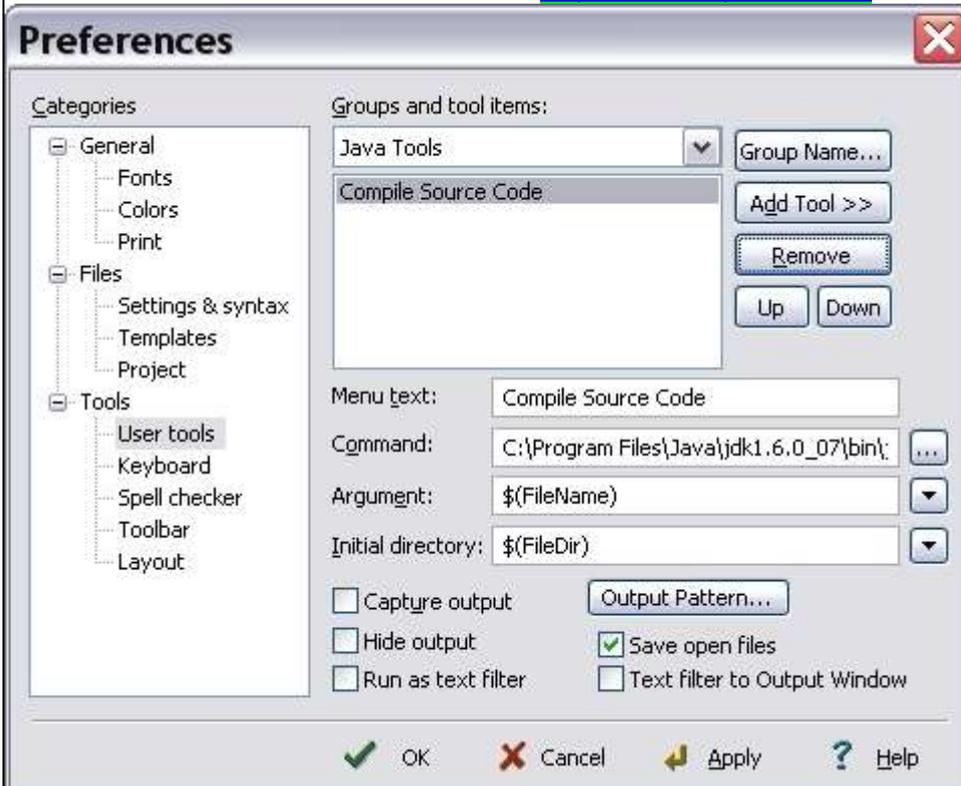
To follow these instructions, you need to know the drive and path where you installed your Java JDK. If you followed the default settings, this is probably `c:\Program Files\Java\jdk1.n.0_xx` (where 1.n.0 is whatever version you installed and _xx is the build number, which may or may not be part of the folder name). If you have a different version of the Java SDK, then the folder name might vary slightly. It doesn't matter what it's called, as long as it contains the "bin" folder with all the JDK utilities.

1. Select "Configure User Tools" from the "Tools" menu.
2. You'll see "Groups and Tool Items" in the middle of the dialog box. Tool groups allow you to organize the different tools you want to create and maintain in EditPlus. We are going to create a tool group called "Java Tools".



- a. You should see a dialog box like the image shown here. The default group name will probably be "Group 1". To change it, click on the **Group Name** button.
 - b. In the New Name text box, type Java Tools
 - c. Click **OK**.
 - d. You should now see "Java Tools" in the pull-down box under the caption "Groups and Tool Items"
3. Now you need to add some tools to your tool group. We'll start with a tool used to compile Java Source code files. If you don't have the Configure User Tools dialog box open, follow step 1 above.
- a. Click the **Add Tool**>>> button and select **Program** from the pop-up menu.
 - b. In the Menu Text field, type "Compile Source Code". This is the text you will see in the menu for this tool.
 - c. Select the program that will run when you select the "Compile Source Code" menu item: Click the small button to the right of the "Command" field.
 - d. Browse to your Java JDK folder and go into the "bin" subfolder. Find and select the file called **JAVAC.EXE**. Make sure you select the right file - there are a lot of files in this folder and they have similar names. When you're sure you have the right file, click Open. You should see the path and file information in the "Command" field. Of course you could type this in by hand instead of using the Browse button if you wanted to.
 - e. When you run the Java compiler, you always give it the name of the source file as an argument. We can set this up for this tool: Click the button to the right of the "Argument" field and select "File Name". You should see \$(FileName) appear in the field.
 - f. Click the button beside the "Initial Directory" field and select **File Directory**. You should see \$(FileDir) appear in the field. This ensures that your compiler runs as if it were in the same folder as your Java source file.
 - g. Click the **Apply** button to save your tool changes.

When finished, your dialog box should look like this one:



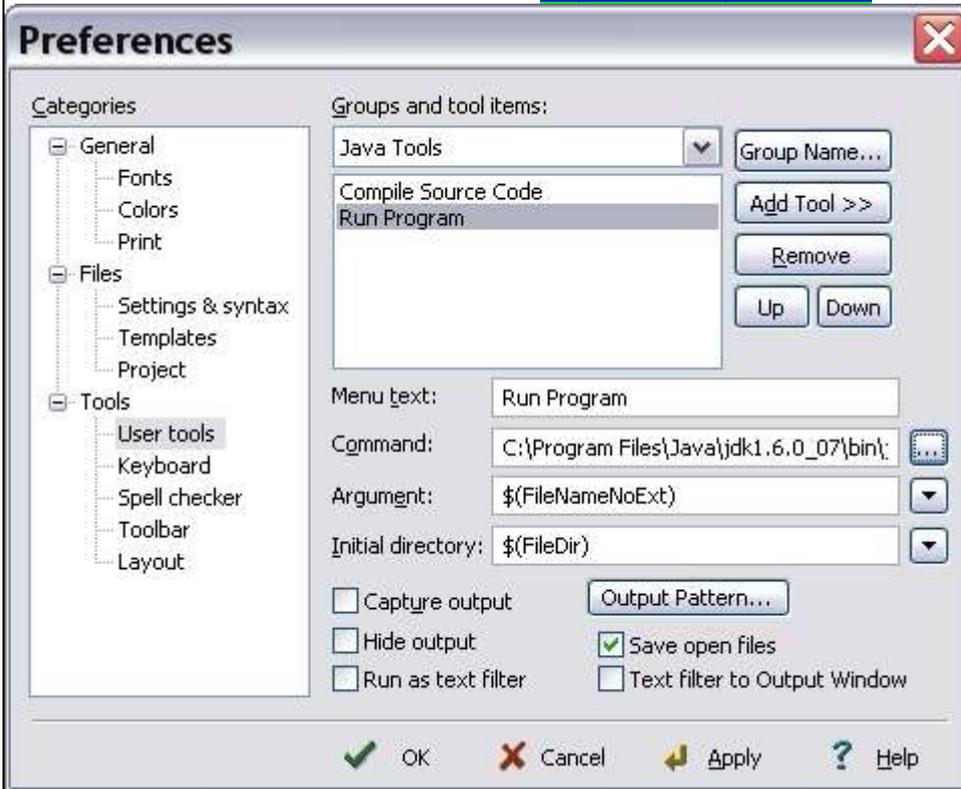
Newer versions of EditPlus have an additional checkbox "Save Open Files". Checking this will cause EditPlus to automatically save your files when you execute the tool. This is handy when you're making changes to a program and want to compile it quickly - you don't have to press the Save button first!

Setting Up the "Run" Tool

The next tool we need to add is for interpreting and running our Java bytecode files. Again, if you aren't in the configure user tools dialog box, follow step 1 above.

1. Click the **Add Tool>>>** button and select **Program** from the pop-up menu.
2. In the Menu Text field, type **Run Program**. This is the text you will see in the menu for this tool.
3. Select the program that will run when you select the "Run Program" menu item: Click the small button to the right of the "Command" field.
4. Browse to your Java JDK's /bin folder and select the file called **JAVA.EXE**. Double check to make sure you have the right file selected, then click Open. You should see the path and file information in the "Command" field.
5. When you use the JAVA command to run a Java class file, you always specify the name of the class file without the extension. We can set this up for this tool: Click the button to the right of the "Argument" field and select "File Name without Extension". You should see \$(FileNameNoExt) appear in the field.
6. Click the button beside the "Initial Directory" field and select File Directory. You should see \$(FileDir) appear in the field. This ensures that the java command interpreter runs as if it were in the same folder as your Java class file.
7. Click the Apply button to save your tool changes.

When finished, your dialog should now look like the one below.



Compiling and Running Programs

To compile a program, go to the EditPlus window with the source file you want to compile and select "Compile Java Source" from the Tools menu. If you prefer using the keyboard, you can also compile the current program by pressing the Ctrl-1 keys.

To run a class file after a successful compile, go to the EditPlus window with the source file you want to run (rather, the source file that is associated with the class file you want to run), and select "Run Java Class" from the Tools menu. If you prefer using the keyboard, you can also run the class file by pressing the Ctrl-2 keys.

Note that the keyboard shortcuts may vary if you have already created user tools. You can see what shortcut keys are associated with your tools by looking at your tool menu items in the Tools menu.

Other Important Notes

Note that sometimes when you compile Java source files and you get errors, the little caret ^ symbol that points to the spot in the code statement may not necessarily be in the right location. This is probably because of the fonts you are using in the EditPlus program (even if you didn't change the fonts). For the most part this shouldn't be a problem since you will probably know exactly where your error is in the offending statement, but if not, you will have to try compiling your program in DOS instead.

Setting Up Other Tools

Depending on your course, your professor may want you to set up some additional tools. These instructions refer to setting up the Run tool to work with command-line arguments, and setting up the AppletViewer tool.

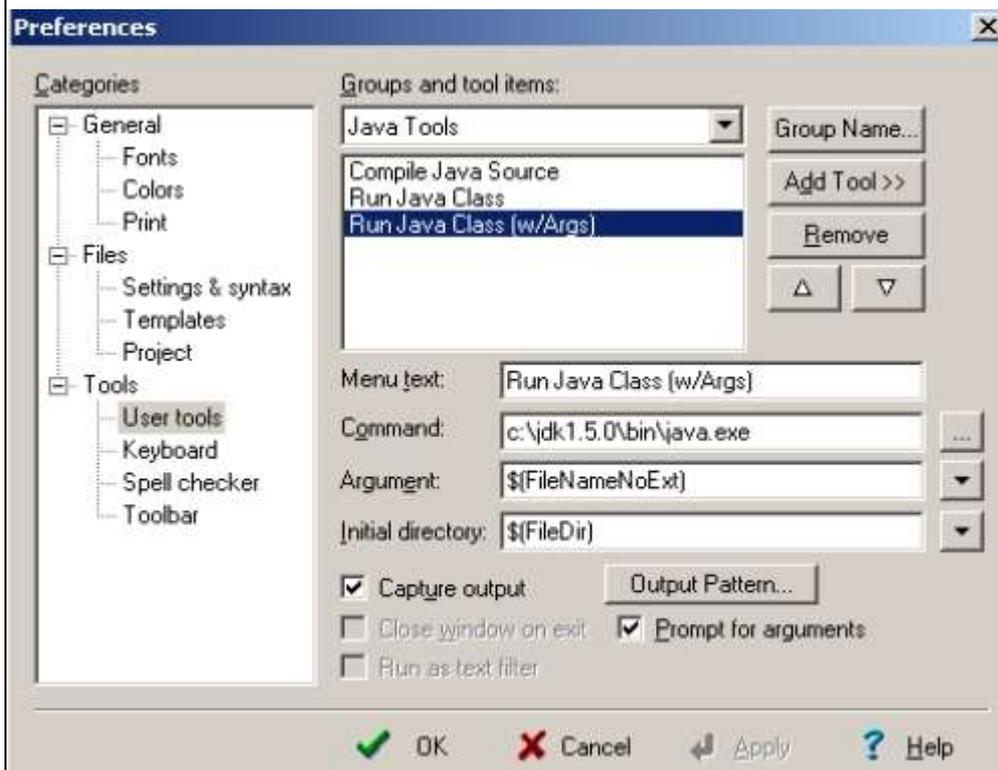
Command-Line Arguments

If you will be using Command-Line Arguments in your course, you will need to set this up, also. If your professor has not mentioned command-line arguments, you do not need to set up this tool. ***Do not set up this tool unless your professor tells you that you need it.***

Older Versions of EditPlus:

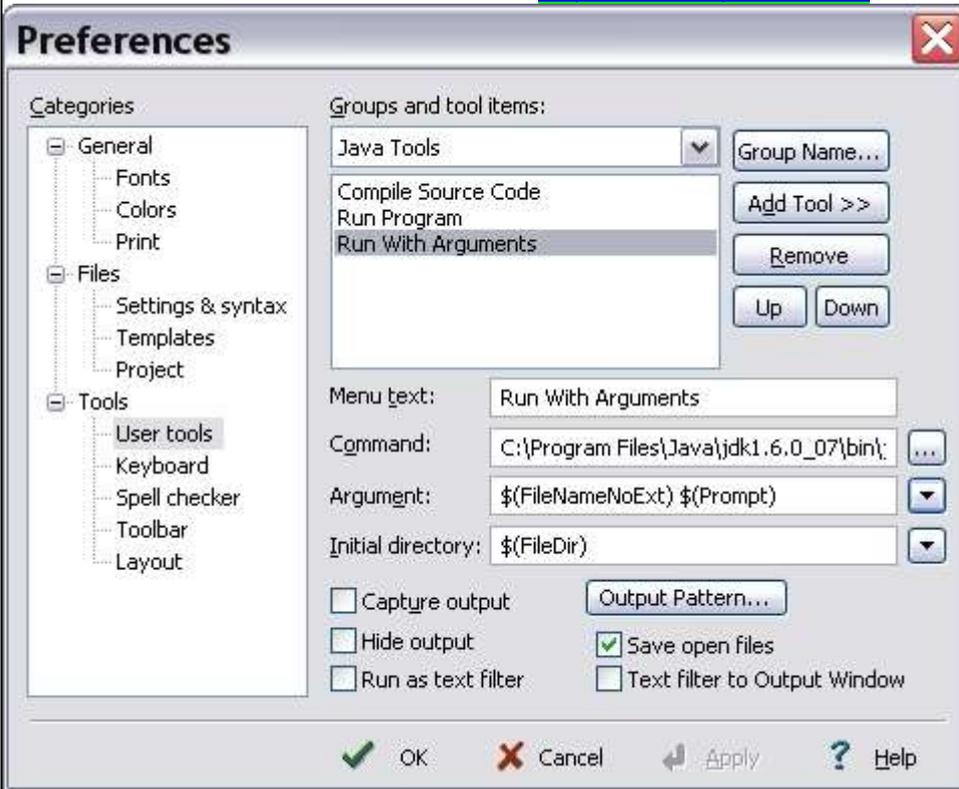
To run a program using command-line arguments, set up a new tool menu just like the one in step 3, but make sure you check off the box labeled "Prompt for Arguments". This will cause a dialog box to appear where you can type your command-line arguments (in the exact same format you would type them at the Dos prompt).

You could also check off this box in the tool menu item you created in step 3, and when you run a program that does not require arguments, click the "Skip" button in the arguments dialog.



Newer Versions of EditPlus:

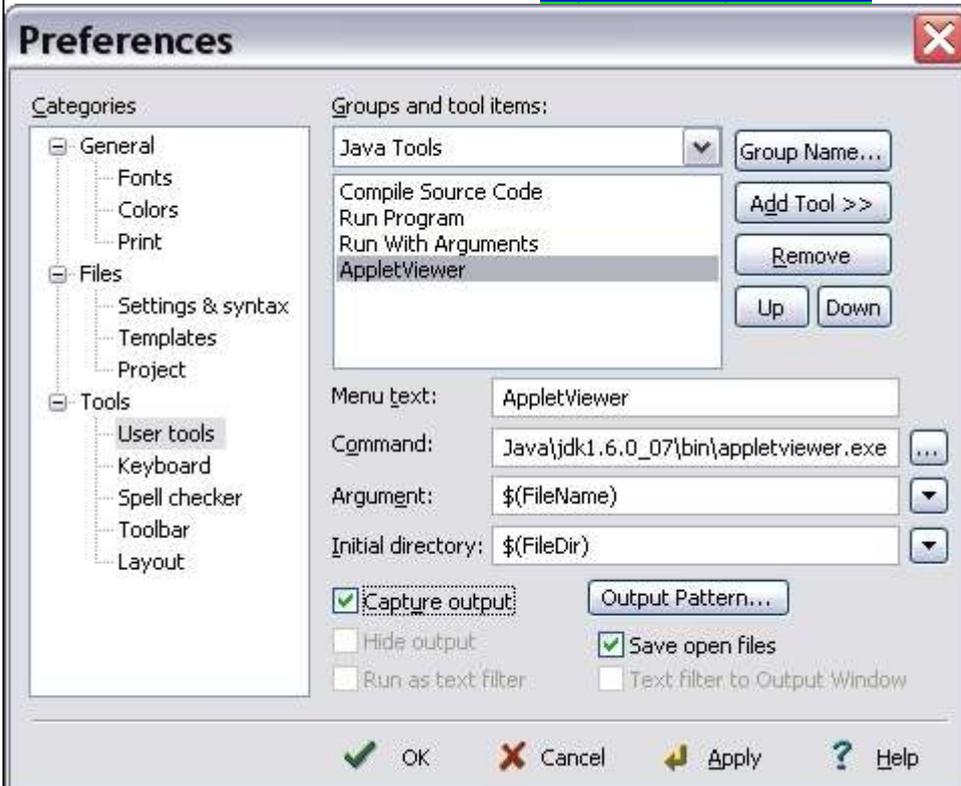
If you don't see the checkbox option mentioned above, you are using a newer version of EditPlus. To set up command-line arguments, Set the "Argument Field": After you select the option "File Name Without Extension", add a space, then add the "Prompt for Arguments" option. Your field should appear as `$(FileNameNoExt) $(Prompt)`, just like the image below.



Setting Up AppletViewer

If you are doing applets in your course, you will need to use AppletViewer. ***Do not set up this tool unless your professor tells you to set it up.***

You can also set up EditPlus to use Appletviewer. The image below shows the settings for the Appletviewer utility:



Dialog box with Run Applet settings (used on an open HTML file!!)

Editing the Auto-completion File

The auto-completion file is the part of EditPlus that is responsible for the automatic completion or adding of elements to various commands you type. For example, if you open up any new or existing Java file and type the words `for` then press the space bar, you'll notice that suddenly all the brackets, braces and semi-colons needed in a `for` loop appear!

Everyone has different preferences when they write Java code. I myself prefer to put my opening brace on the same line as the statement that begins a block of code, for example:

```
public class Temperature {
```

Unfortunately EditPlus likes the brace on the next line by itself. Luckily we can change this if we want!

In the Program Files/EditPlus 2 folder is a file called **CPP.ACP** that contains all the "syntax" for automatically completing bits of code in Java and C++. You can edit this file by opening it in EditPlus and making the changes you like (such as moving the opening brace to the end of the line above!).

You'll notice that the characters `^!` appear throughout the file. This indicates where your insertion point will end up after the code appears. You can move that, too!

Be careful when editing this file: Don't change lines that start with a `#` character or `;` character.

Updating the Syntax File

When you type Java programs, you'll notice that some of the words turn blue, some pink, some green, some red, etc. That's because of the syntax file EditPlus associates with Java programs. Usually comments are green, classes are red, keywords are blue, and so on. The version of EditPlus on your computer has an older syntax file. You might want to update to a newer file that was created for Java 1.5.0. You don't have to

update this file - it won't affect how your programs compile and run. You may want to update it however, so that some of the newer classes added to later versions of Java show up in red text when you code them. The older syntax file doesn't contain entries for classes like JOptionPane, so it won't turn red when you type it. Having an up-to-date syntax file is helpful for programmers new to Java - if the class name doesn't turn red, you know you've typed it incorrectly.

To update your syntax file:

1. Make a backup of your current syntax file:
 - a. Go to c:\Program Files\EditPlus 2 and locate the java.stx file (your location might be different, depending on your installation).
 - b. Rename the java.stx file to java.stx.bak.
2. Download the [javasyntax.zip](#) file.
3. Unzip this file into the c:\Program Files\EditPlus 2 folder.

If you want to download and add more syntax files and other files, check out the [EditPlus User Files](#) page.

Editing the Template File

When you open a new Java program, you'll get some default code, such as:

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

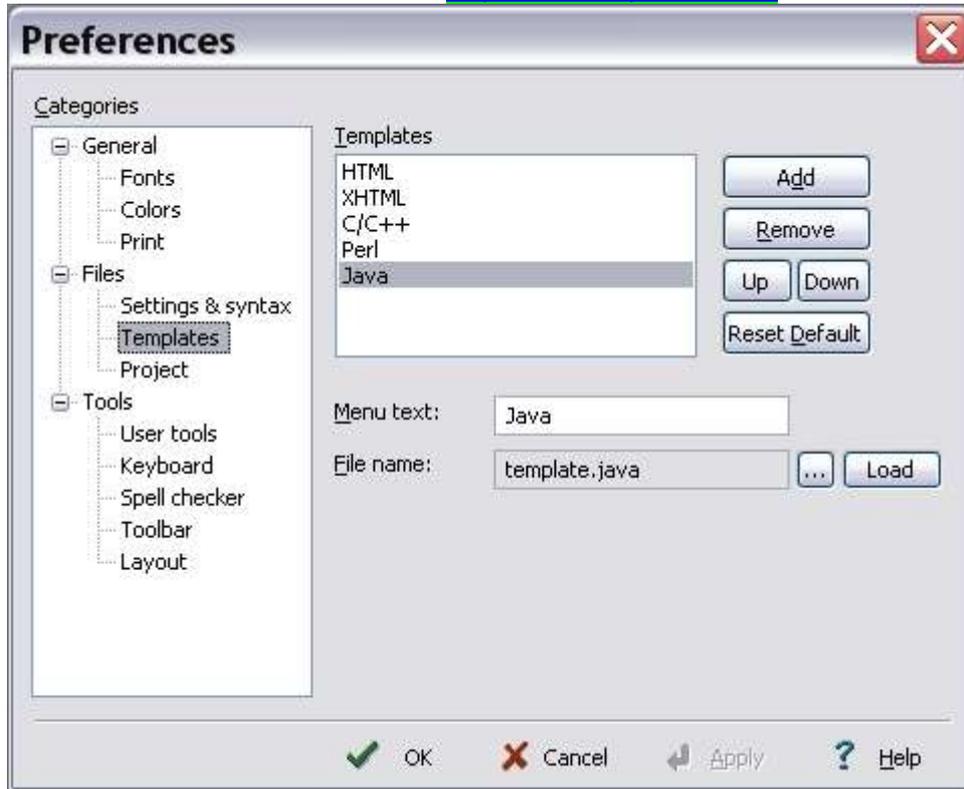
We don't want the Hello World statement in there every time we start a new Java program, and we also want to make sure that all of our classes are public by default (otherwise you'll forget to add the word "public" yourself!).

Another handy thing would be if you had the proper comments that identified your program at the top of the code! As you know (or will soon learn) all of your programs must be properly identified with your name, class, date, and assignment information.

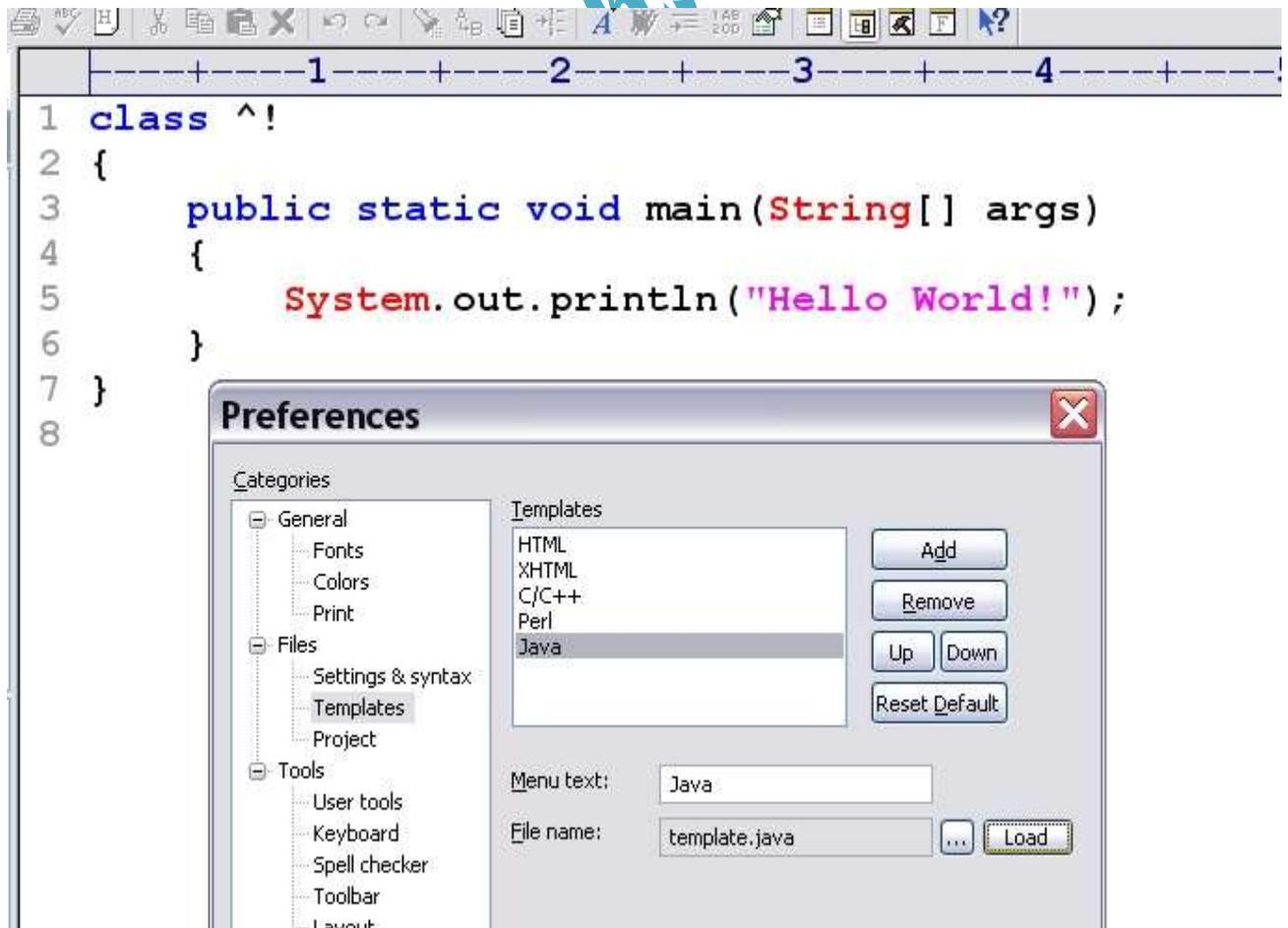
You can arrange it so that every new Java file you start has the proper program identification, the word public in front of the keyword "class", and no Hello World statement by editing the Java template file. To edit this file, do the following:

NOTE: This only works with newer versions of EditPlus. If you have an older version and the steps below don't work or aren't applicable, contact your professor.

1. In the EditPlus window, choose the **Tools** menu, then select **Preferences**
2. In the preferences window, open the **File category** (if it's not already open) and then select **Templates**.



3. To load the current template, click the **Load** button.
4. You should see the template load in the background just like in the image below:



5. Click the **OK** button to get rid of the Preferences dialog.

You can now add and change the existing template to your liking. You'll need to do the following to your template:

- Add the keyword `public` (followed by a space) in front of the word "class" in the class header.
- Add the necessary programmer documentation to the top, as per the <http://sbinesh.yolasite.com>
- Delete the line that prints "Hello World!" on the screen.

For example, a good `template.java` file for a student named Jen MacDonald in the Computer Science program might be:

```
/**
 *      Name      : Binesh Singh Rajput
 *      Email     : mcabinesh@live.com
 *      Web       : http://bineshsingh.yolasite.com
 *      Date      : [due date or date of example]
 *      @author   : Binesh
 */

public class
{
    public static void main(String[] args)
    {
        System.out.println("Hello Binesh!");
    }
}
```

When you've edited your template file, save the changes by pressing the **Save button**.

NOTE: The template code will not appear in any existing Java files, only in the new ones you create from now on.

Note: If u like then must visit <http://sbinesh.yolasite.com> and also give your valuable feedback for further updation.

Thanks & Regards

Binesh