

---

# How to Run Java Programs Using Symantec Visual Cafe

---

**C. Thomas Wu**

**Supplement material to accompany *An Introduction to Object-Oriented Programming with Java***

*WCB/McGraw-Hill*  
© *The McGraw-Hill Companies, Inc.*, 1999

---

In this document, we will explain how to run Java programs using Symantec Visual Cafe version 3.0 (VCafe). Please remember that the procedure described in this document is just one way to create and run Java programs with VCafe. Among the many possible variations, we present here the most straightforward.

---

## **1.0 Installing the Software and javabook Package**

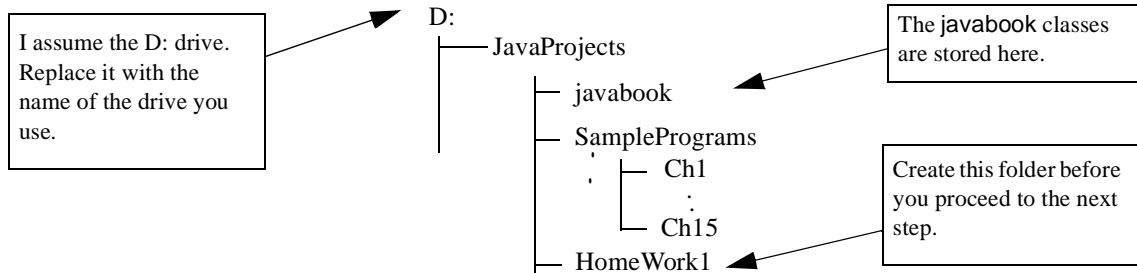
---

Install VCafe on your machine following the instructions of the installation program. Also, download the javabook package and sample programs from the DrCaffeine web site ([www.drcaffeine.com](http://www.drcaffeine.com)). Notice that the Java package is organized as a folder (directory), so you will see the javabook package as a folder that contains the javabook classes. The javabook folder also includes the source files so you can study them and have an option of modifying them.

## 2.0 Directory Structure

---

You can place the `javabook` package and sample programs anywhere you want. However, in order to present concrete examples on how to run programs that use `javabook` classes, we will assume the following directory structure in this handout:



Although you can create project folders such as `HomeWork1`, we suggest you to create the project folders before working with VCaFe.

## 3.0 Starting VCaFe

---

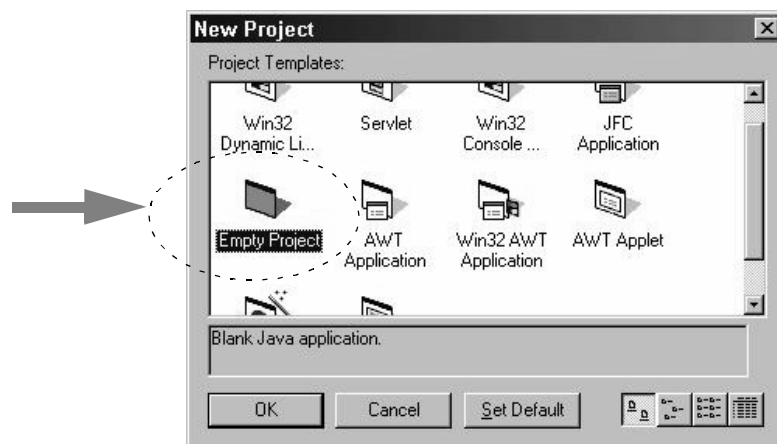
Start VCaFe by selecting its program icon in the `Start|Program|Visual Visual Cafe` group (or in whichever group the VCaFe icon is located).

## 4.0 Creating a Project

---

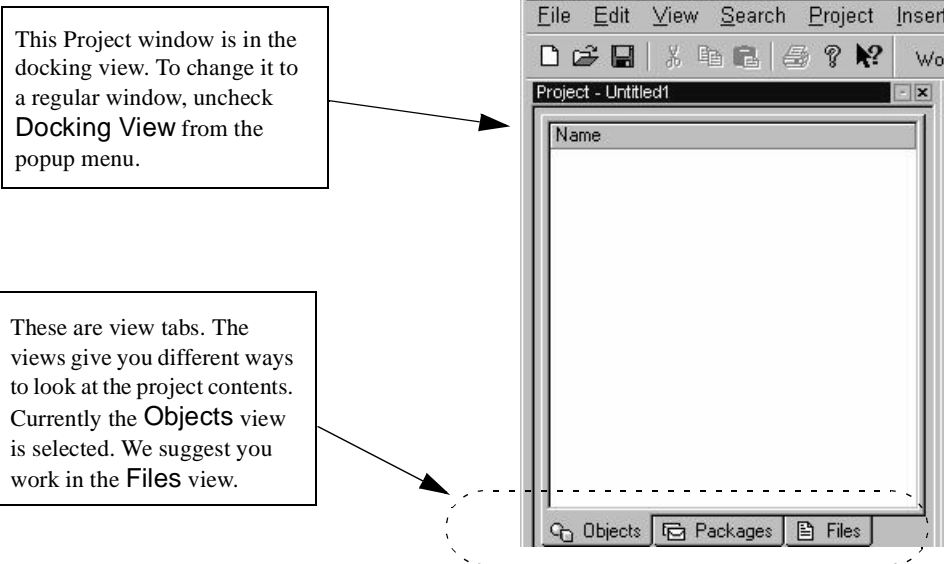
VCaFe and other Java IDEs (Integrated Development Environment) use project-based concepts. A project consists of various types of files that are necessary for compiling and running programs. For the sample programs from the earlier chapters, the projects will include only one or two source files (note: we are counting here only the files we create ourselves). In the later chapters, projects will include over 10 files.

To create a project, select the menu choice `File|New Project`. The New Project dialog will appear on the screen.



Click the **Empty Project** icon. We are creating a project that does not include any template files. If you select other types of project, you will get template files that provide some basic functionality (such as showing a menu). These template files are convenient if you have a certain level of Java programming skills. Since we need to learn from the very beginning, we start with an empty project without using any of these template files.

After the **Empty Project** icon is selected, click the **OK** button. The Project window will appear:



The Project window is displayed in the **docking view** (the window is “docked” to the VCafe window). You can display it as a regular window by unchecking the menu item **Docking View** from the popup menu. Use whichever view you prefer.

---

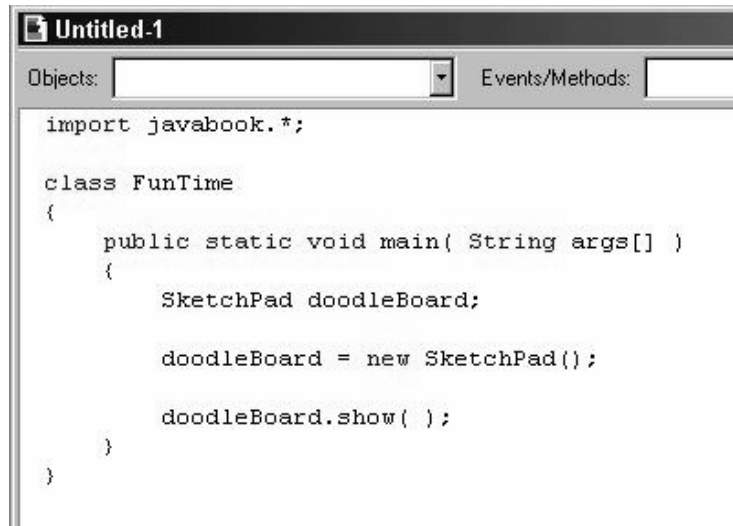
## 5.0 Creating a Class File (Source Code)

---

We are now ready to add a class to the project. Let's create the **FunTime** program from Chapter 1. Choose the **File|New File** menu choice. The Editor window will appear on the screen.



Type in the FunTime program. After you entered the program, the editor will look something like this:



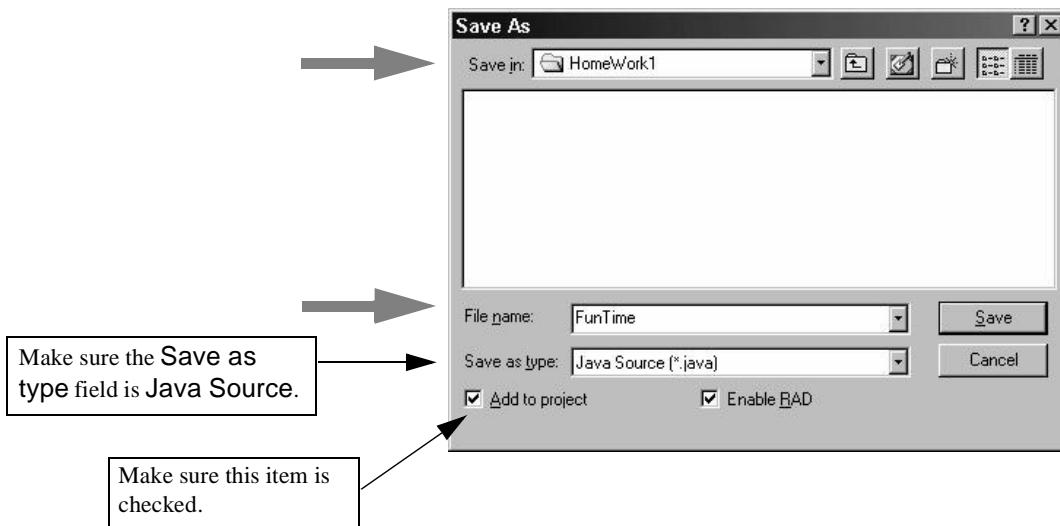
```
import javabook.*;

class FunTime
{
    public static void main( String args[] )
    {
        SketchPad doodleBoard;

        doodleBoard = new SketchPad();

        doodleBoard.show( );
    }
}
```

After the program is entered, chose the File|Save As menu choice to save the program. The Save As file dialog will appear on the screen. Navigate to the directory HomeWork1 (see Section 2.0), and save the source file as FunTime.java. Make sure the CheckBox item Add to Project is checked. **NOTE: You don't have to enter the suffix .java. It will be added automatically if Java Source (.java) is chosen as Save as type.** The Save As file dialog should now look like this:

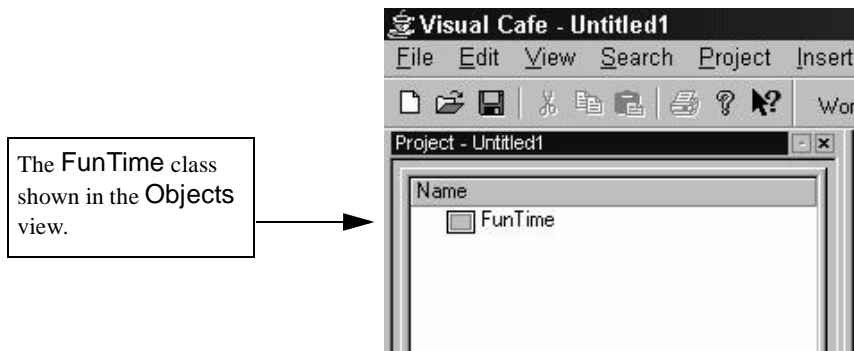


---

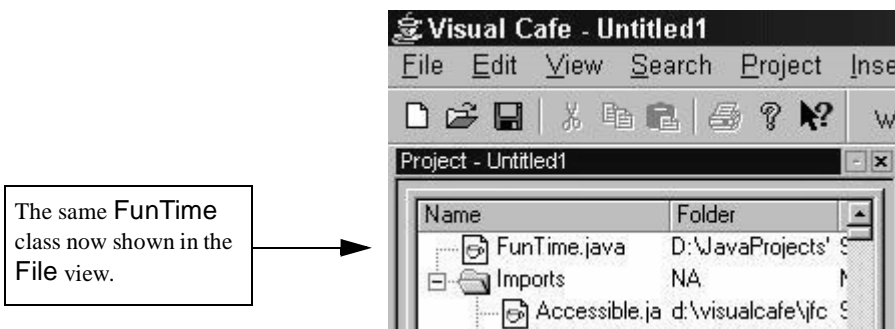
## Specifying the Project Properties (Very IMPORTANT!)

---

Click the **Save** button when everything is set correctly. The Project window will now look like this:



This is the **Objects** view of the Project window. The following is the same Project window in the **File** view. You change to the File view by clicking on the File tab. We recommend you use the File view.

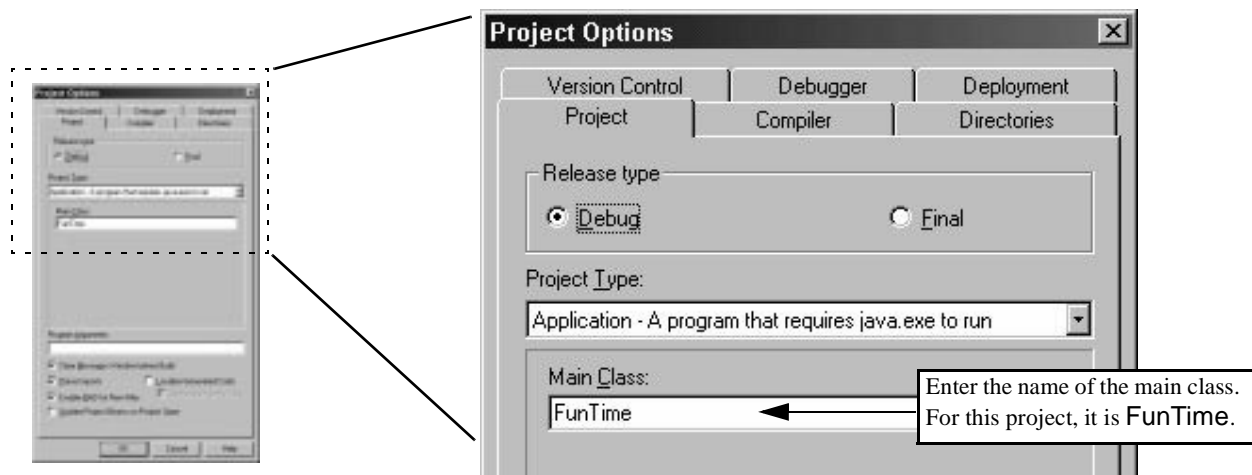


---

## 6.0 Specifying the Project Properties (Very IMPORTANT!)

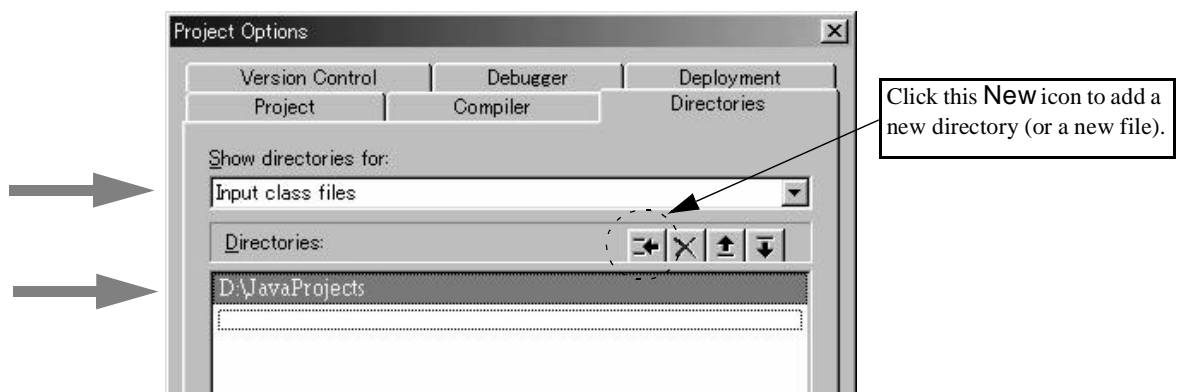
---

Before we can run the FunTime program, we must set certain properties of the project. Choose the menu choice **Project|Options**. The Project Options dialog will appear:



In the **Main Class** field, enter the name of the main class, which is **FunTime** for this project.

Now, click the **Directories** tab. Verify that **Input class files** is shown for the **Show directories for:** dropdown list. If not, select it from the dropdown list. We will add the directory that contains the **javabook** package to the list, so the **FunTime** program can correctly refer to the class in the **javabook** package. To add a new directory (or a new file) to the list, click the **New** icon (see the diagram below). An entry to add the directory (or a file) is added to the list. Type in the directory that contains the **javabook** directory, which, in our case, is **D:\JavaProjects** (see Section 2.0). The **Project Options** dialog now looks as follows:



Click the **OK** button to set the properties and close the dialog.

---

## 7.0 Running a Java Application Program

---

We are finally ready to run the program. Choose the menu choice **Project|Execute**. If there's no error in the program, the **SketchPad** window will appear on the screen.



---

## 8.0 Save the Project

---

Interestingly V Cafe does not have a menu choice **Save Project**. Select the **Close Project** menu item. V Cafe will ask you to save the project or not. Click **Yes**, and save the project under the directory **HomeWork1**. If you need to continue working on the project, you have to open the project by selecting the menu choice **Open Project**.

---

## 9.0 Quick Summary

---

Here's a quick summary of the steps you take to enter, compile, and run a Java application program using V Cafe.

1. Create an empty project.
2. Create a new Java source file. Name the source file with the class name. Example: The class **Funtime** is saved in the source file **Funtime.java**.
3. Enter the name of the main class in the project option's **Main Class** field. Add the directory that contains the **javabook** package to the list of **Input class files**. *Note: You specify the directory that contains the **javabook** package, not the **javabook** package itself. Example: If the **javabook** package is located in **D:\JavaProjects**, then you set the directory to **D:\JavaProjects**, NOT to **D:\JavaProjects\javabook**.*
4. Run the program.

---

## 10.0 Running Applets

---

Steps for compiling and executing applets are similar to those for applications. The main difference is that an applet requires an HTML file. The role of HTML files is explained in Chapter 2. We will outline the steps in a summary format.

1. Create an empty project. To avoid conflicts, create a new directory for each project. Let's say we have a new directory **MyApplet** under **D:\JavaProjects**.
2. Create a new java source file and an HTML file. Name the source file and the HTML file with the class name. Example: The source file for **MyFirstApplet** is saved as **MyFirstApplet.java**, and the corresponding HTML file is saved as **MyFirstApplet.html** (or **MyFirstApplet.htm**). Please refer to Chapter 2.

Save the files in the folder **D:\JavaProjects\MyApplet**. Make sure the **Add to project** checkbox is checked, so the two files are added to the project when they are saved.

3. Run the applet.

*Note 1: You don't have to adjust the project options. If you open the **Project Options** dialog, you will notice the **Project Type** is set to **Applet** and the **Start with Web Page** is set to the HTML file of the project.*

*Note 2: When you run an applet for the very first time, you may get a dialog asking you to accept the Sun license agreement or not. Click the **Accept** button.*

*Note 3: You don't add the **javabook** classes to an applet project unless, of course, your applet uses non-user interface **javabook** classes such as **Convert** and **Format**.*

---

## 11.0 Running the Sample Programs

---

You can run the sample programs from the textbook easily by creating a project. Create a folder (directory) and a project inside this folder. Let's say we name both the folder and the project **Sample**. To run a sample program,

1. If the project contains any files, delete them. To delete files from the Project window, select the files and choose the menu choice **Edit|Delete**. Use any standard technique to select multiple files. To avoid any potential conflicts, you should also delete the files from the folder. Note: Deleting a file from a project does not delete the file from a folder.
2. Copy the relevant files from the sample program folder to the **Sample** folder.
3. Add these files to the project by selecting the menu choice **Insert Files** from the popup menu of the Project window.
4. Change the settings in the **Project Options** dialog accordingly. If the sample program is an application, set the **Project Type** to **Application** and enter the name of the main class in the **Main Class** field. If the sample program is an applet, set the **Project Type** to **Applet** and enter the name of the HTML file in the **Start as Web Page** field.
5. Run the program.