

Programming in Two Semesters

Charatan and Kans

Installing and Using Apache NetBeans™

1. Installing Apache NetBeans™

Before downloading and installing the Apache NetBeans™ tool ensure that you have downloaded and installed the Java SE Development Kit (JDK) from the Oracle™ site. For novice users, it is recommended that you download JDK version 8, which means you will be able to run JavaFX programs (chapters 23 and 24) without having to further configure NetBeans:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

If, however, you wish to install the latest release, you can do so from the link below. However, in order to then run JavaFX programs (chapters 23 and 24), you will need to further configure Netbeans™ as described in the accompanying document (JavaFX instructions for JDK 11 and later).

<https://www.oracle.com/technetwork/java/javase/downloads/>

Once you have downloaded and installed the JDK you can then download and install the Apache NetBeans™ IDE from the following site:

<https://netbeans.apache.org/download/index.html>

2. Using NetBeans™

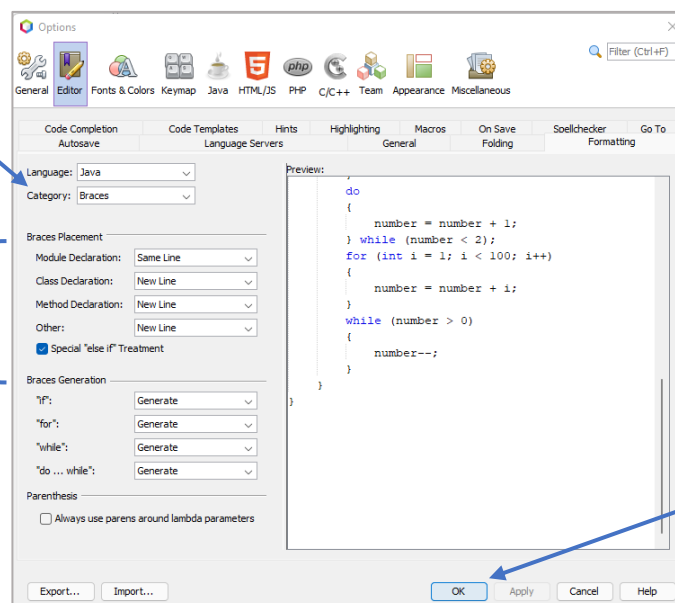
2.1 Configuring NetBeans™

Apache NetBeans™ comes preconfigured with various settings that can be customised if required. For example, to follow the curly bracket style of the text (with each bracket on a new line) follow these instructions:

- Navigate to **Tools -> Options**
- Navigate to **Editor -> Formatting** and set the **Braces** options as shown in figure 1

Select **Java** from the Language list and **Braces** from the Category list

Select **New Line** for each Braces placement option



click **OK** once to complete

Figure 1

2.2 Starting a new project

All the files associated with an application should be kept in a single project. It is recommended that you create a separate project for each application that you are creating. To create a project, chose *New Project* as shown in figure 2.

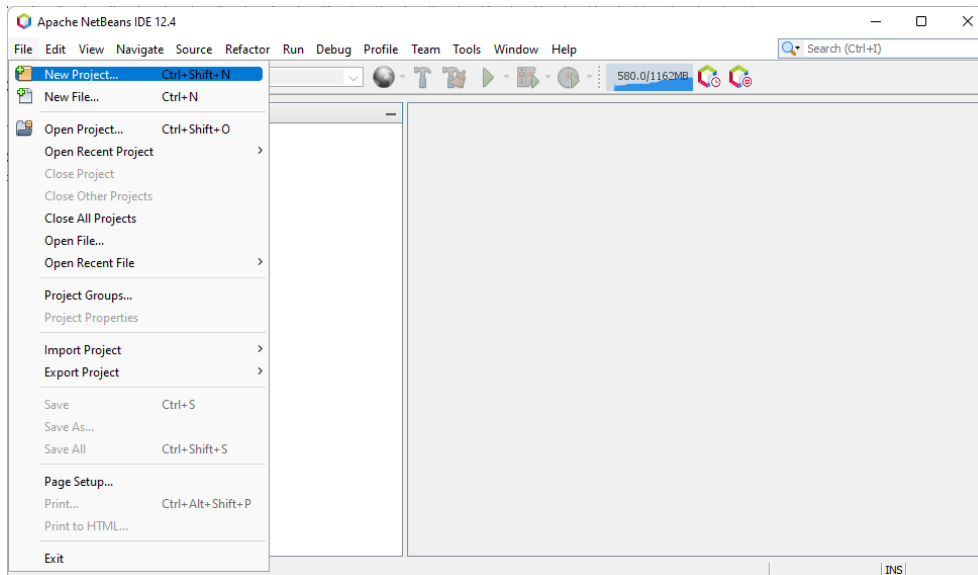


Figure 2

You will now see the screen shown in figure 3. From the *Categories* window, select *Java with Ant*, and then choose *Java Application* from the *Projects* window. Then press *Next*. Leave the *Filter* field blank.

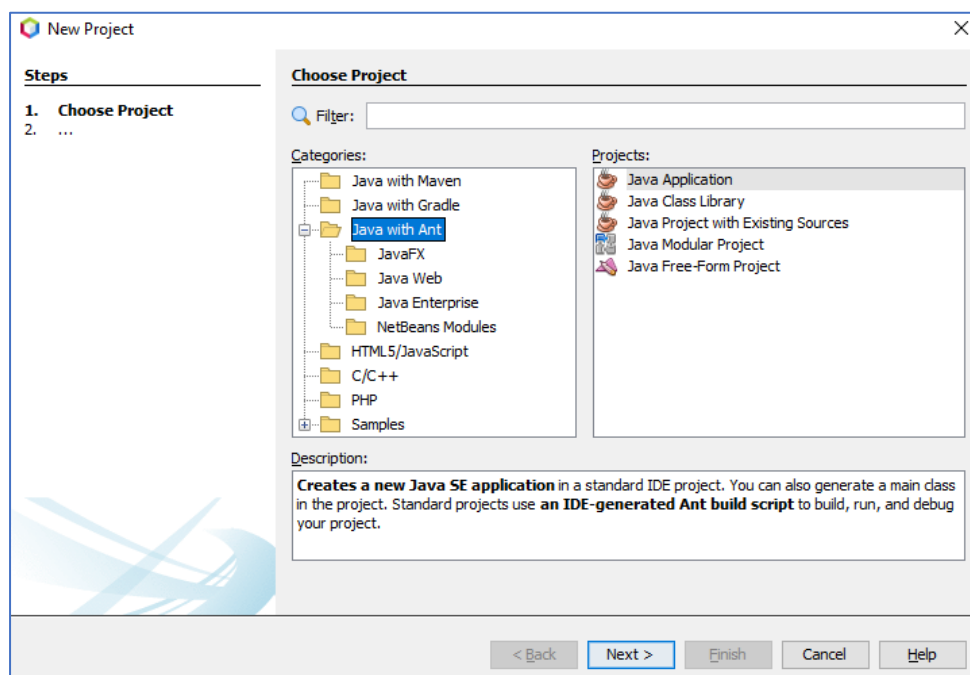


Figure 3

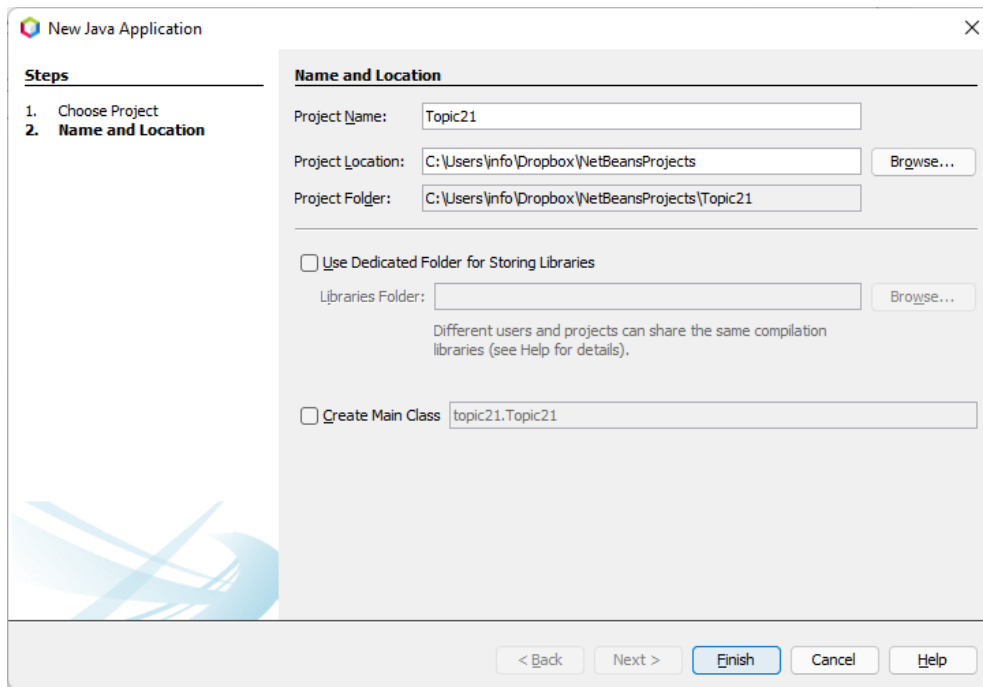


Figure 4

Enter the name of the project (in the above example *Topic21*) in the *Project Name* field.

In your early programs you are not going to create packages for your applications, so you want all your source files to be kept in what NetBeans™ calls a **default package**. The easiest way to achieve this is to uncheck the *Create Main Class* box, and create your main class later. Now you can press *Finish*. Your screen will now appear as in figure 5, with *Topic1* listed as your only project.

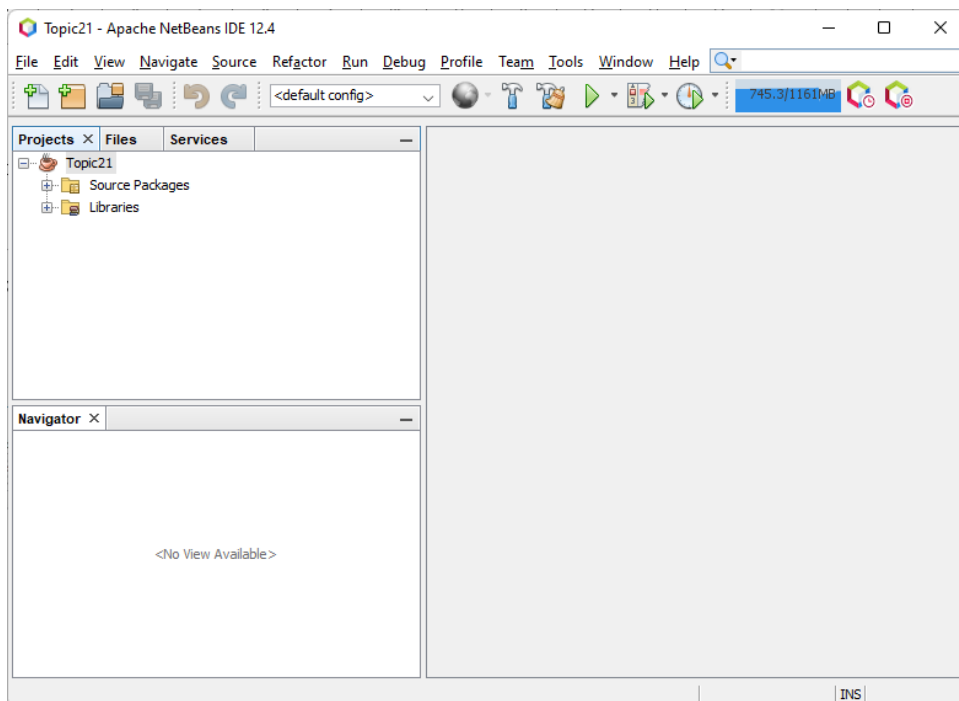


Figure 5

2.3 Adding new files to the project

To add a new file to your project, you can either choose *New File* from the *File* menu, press the + sign in the top left hand corner, or type Ctrl+N. You will see the screen shown in figure 6.

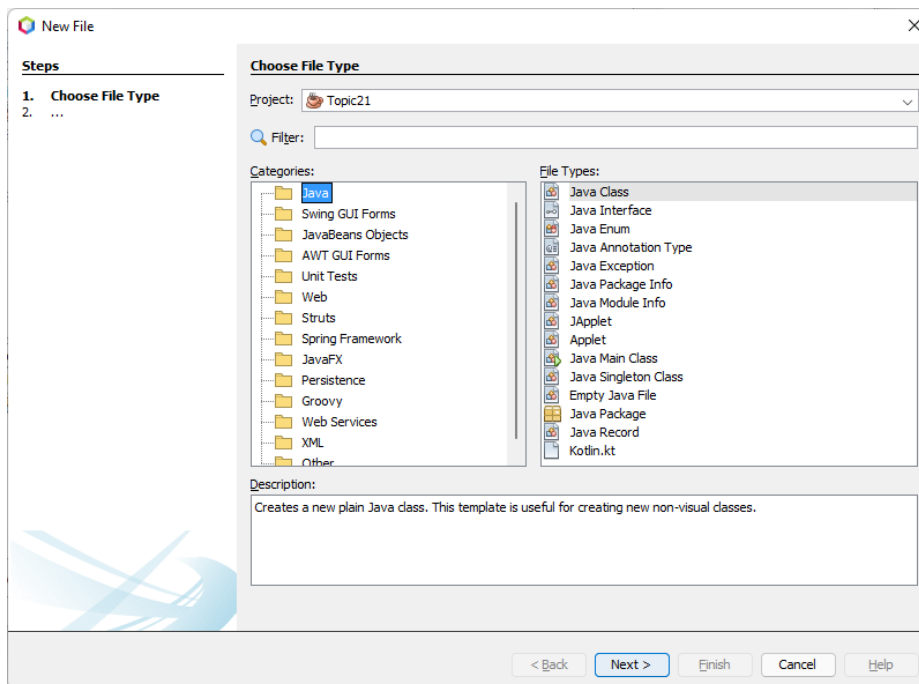


Figure 6

Choose *Java* from the categories section, and from the *File Types* window you can choose *Java Class* (provides just the class header, and opening and closing braces), *Java Main Class* (provides an outline main method) or *Empty Java File*, depending on which you want.

Pressing *Next* will cause a new screen to appear, as in figure 7.

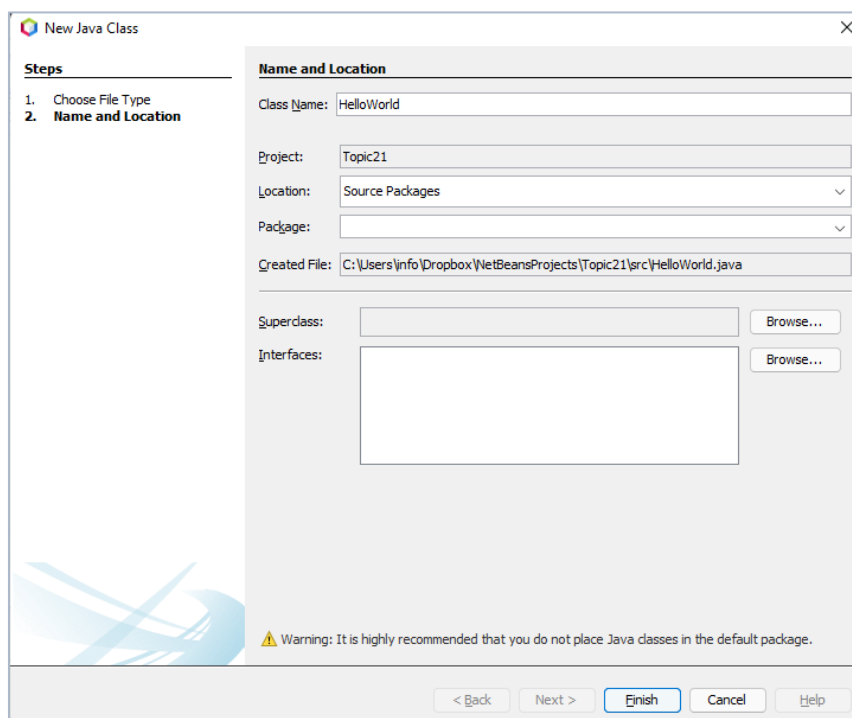


Figure 7

Type in the name that wish to give to your class (*HelloWorld* in this example), and press *Finish*. The class will be listed in the default package, and (if you had chosen *Java Main Class*) the code will appear as shown in figure 8. If you had chosen an empty class, this would be blank.

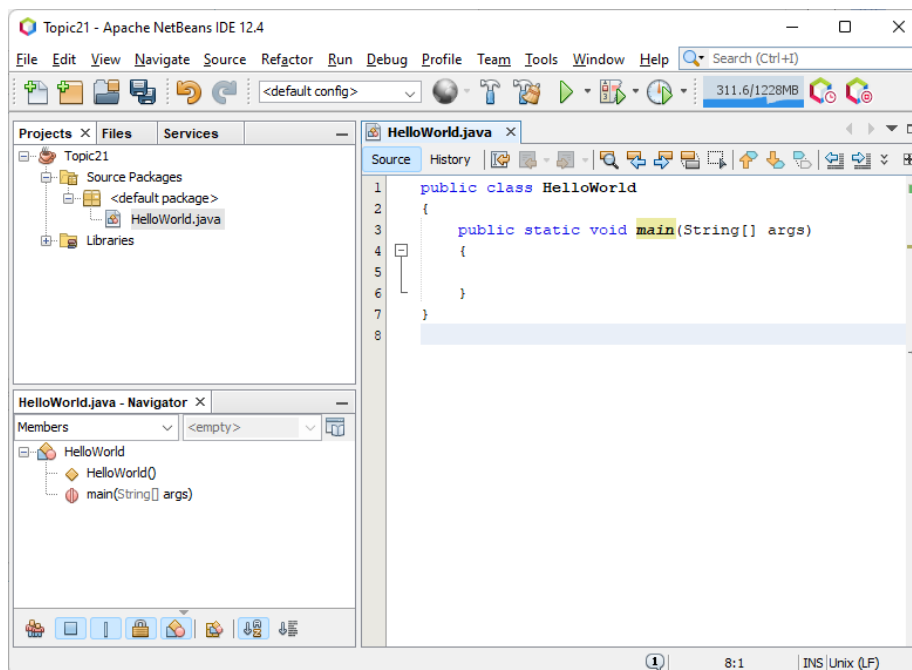


Figure 8

2.4 Compiling and running applications

Classes are automatically compiled as soon as they are written. Error messages are indicated in red and are revealed when you hover over the red icon with the mouse (as shown in figure 9).

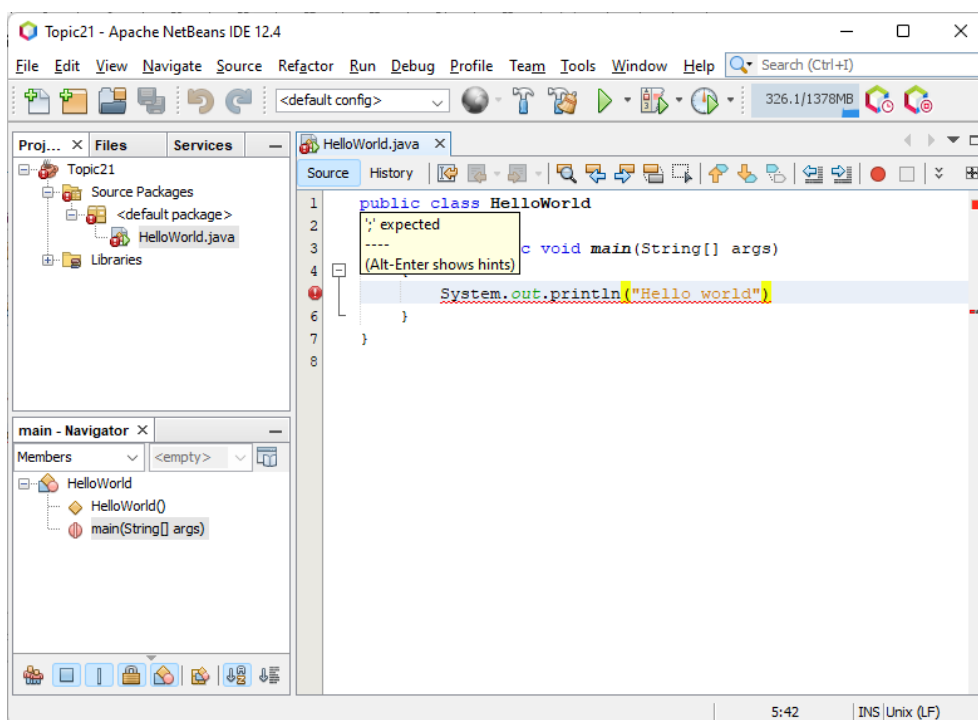


Figure 9

Programs can be run by pressing shift+F6 on the active window containing the main class; alternatively you can right-click on the class in the list and choose *Run File* from the drop down menu (or again press shift+F6).

Running a non-graphics program will cause the output to be displayed in an output window as shown in figure 10. This window can be made to float, if you prefer (this is most easily done simply by dragging it).

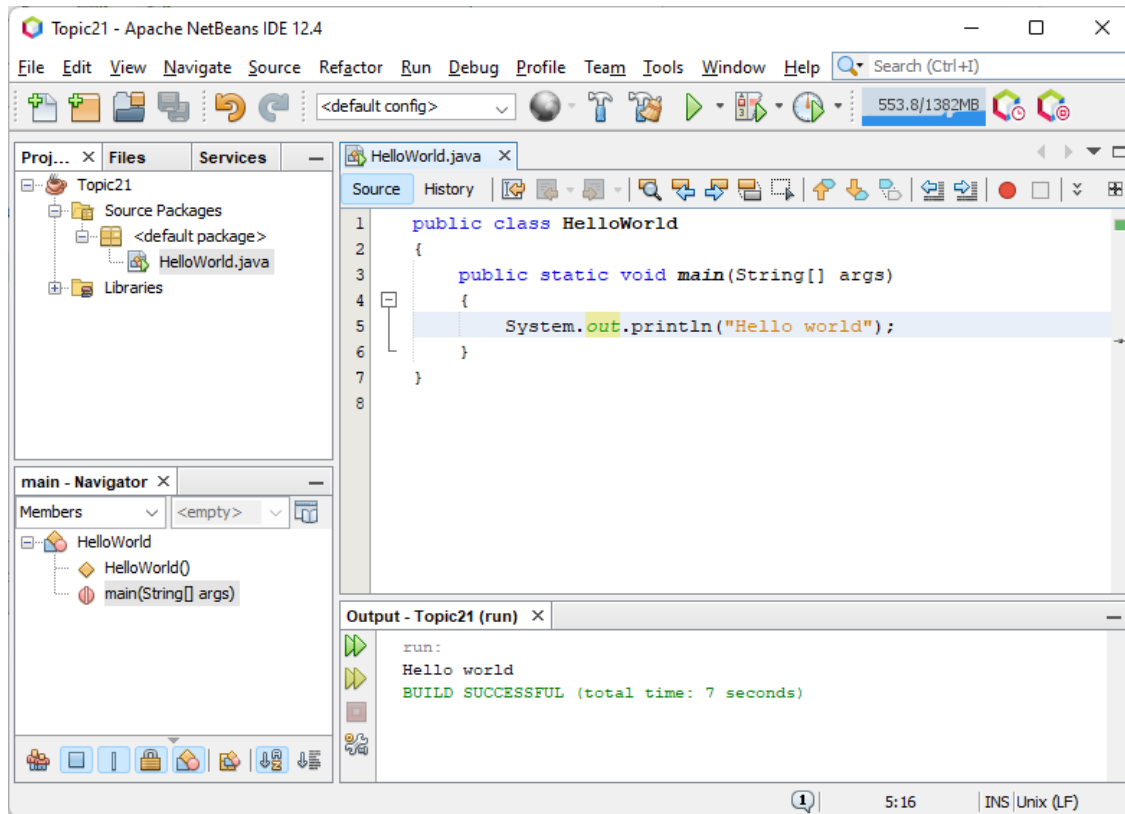


Figure 10

2.5 Adding existing classes to projects

Existing classes (that is to say, the source code) can be added to the project simply by placing them in the correct folder.

For example, if the default directory for your NetBeans projects is called *NetBeansProjects*, and your project is called *Topic1*, then the *.java* files are kept in a folder called *src*, as shown in figure 11. If you had created your own package, then files residing in that package would appear in a subfolder of *src* with the package name, as shown in figure 11.

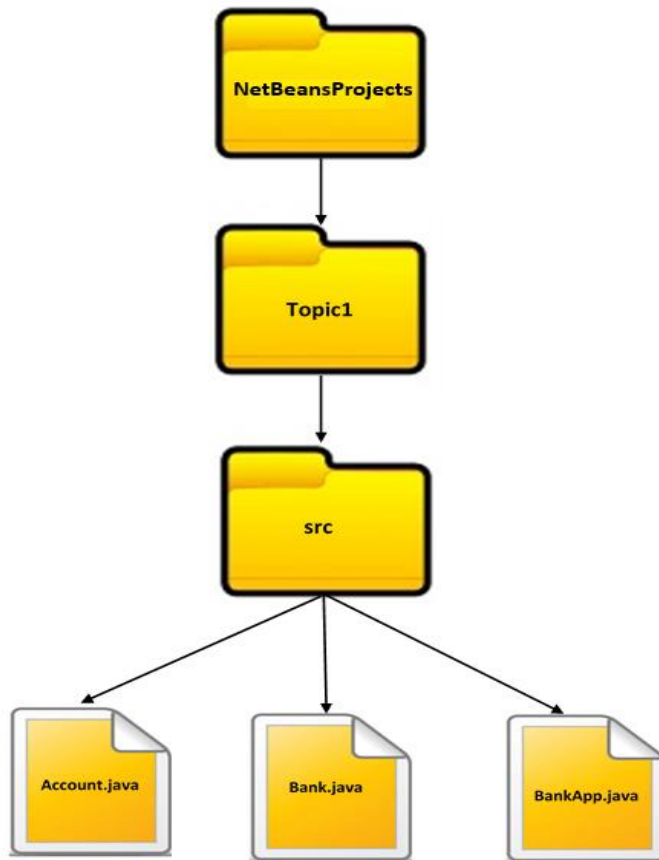


Figure 11

Alternatively you can drag the required file into the package while NetBeans is open (for example in figure 12, drag it into the *<default package>*).

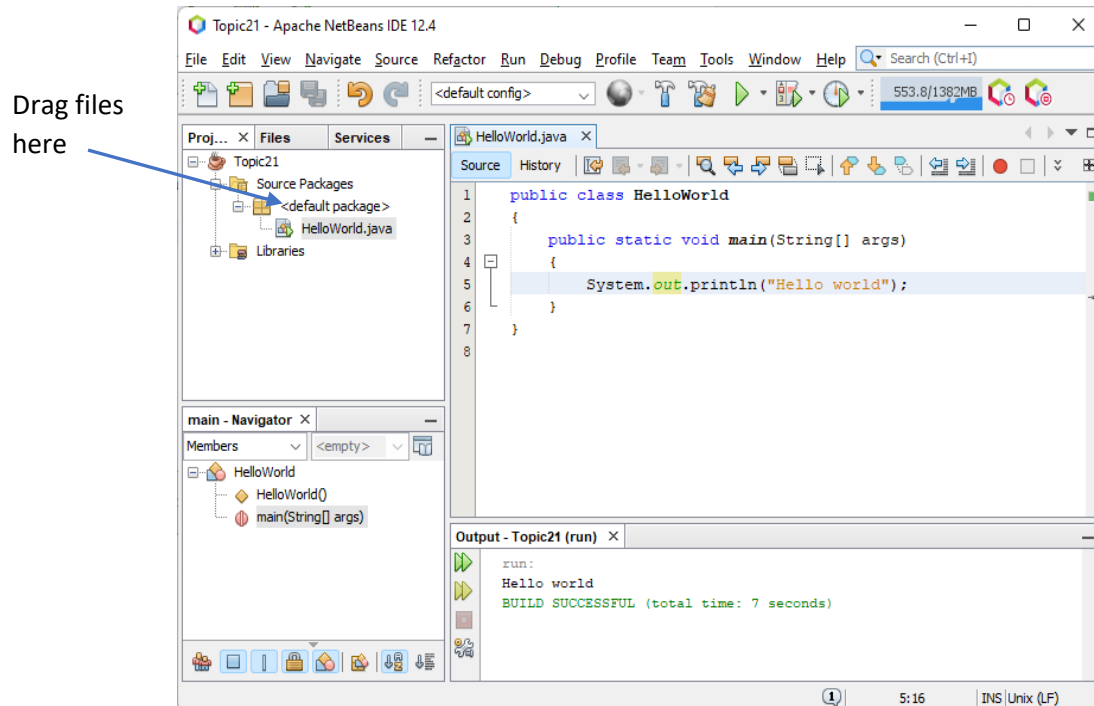


Figure 12