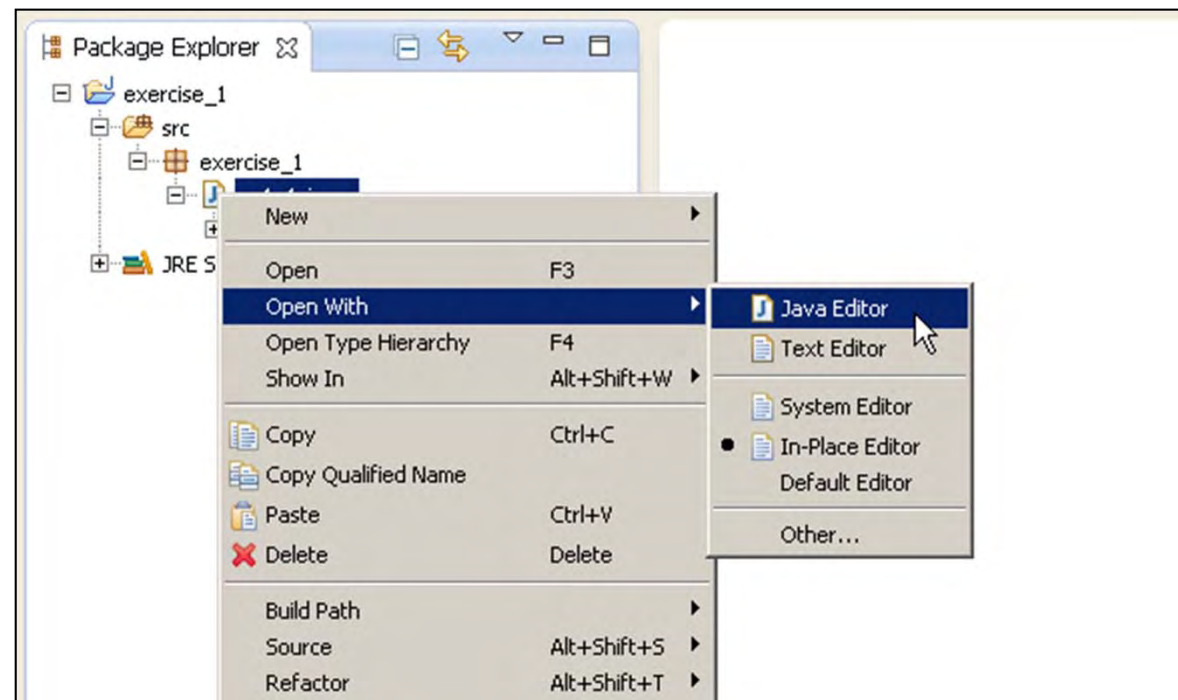


Using the Java Editor – Introduction



One of the main advantages when using Eclipse, is its very comprehensive hint and error handling within the Java Editor, this section breaks down the usage of the Editor.

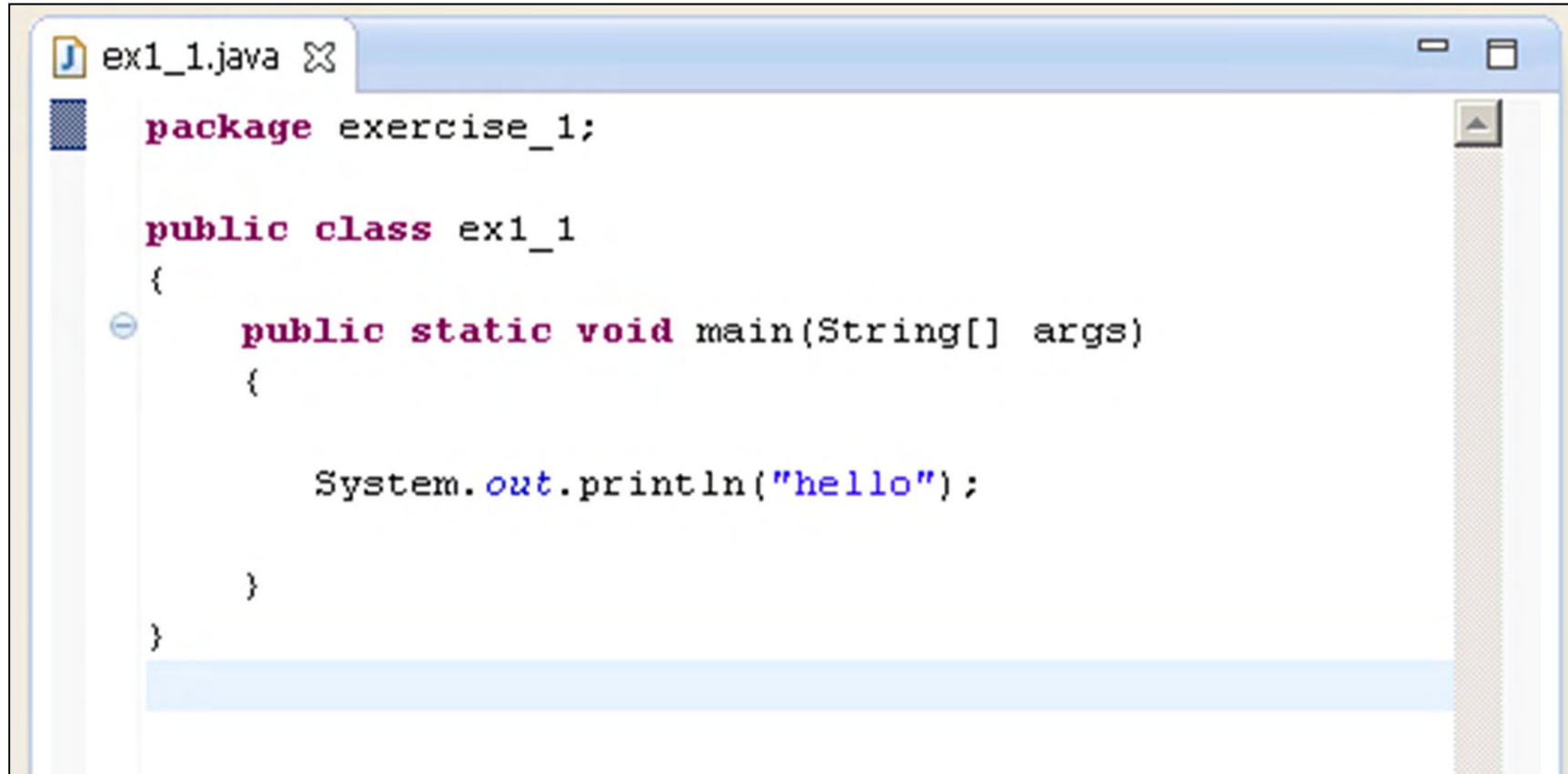
By default double clicking on the class or method will open up the Java Editor, however a right click will allow the Developer to use *'Open with' ...*



Using the Java Editor – Introduction



The Java Editor will open the selected Java in a one of its tabs ...



```
ex1_1.java X
package exercise_1;

public class ex1_1
{
    public static void main(String[] args)
    {
        System.out.println("hello");
    }
}
```

Using the Java Editor – Folding



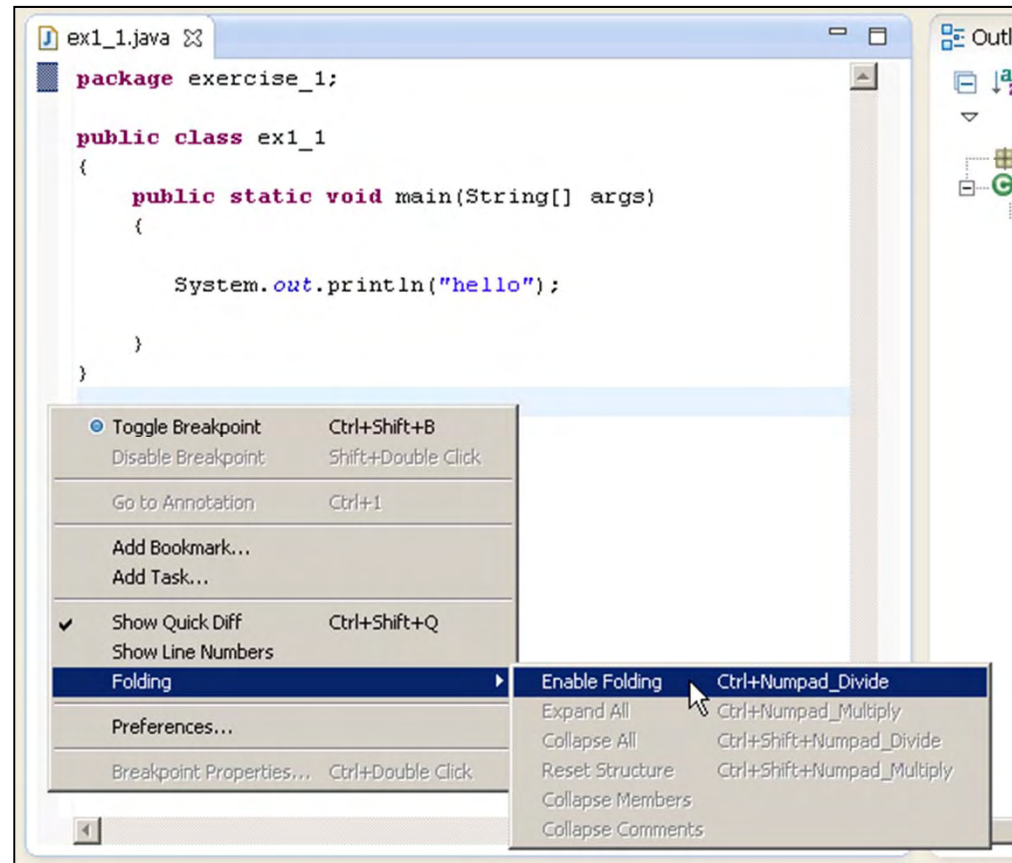
Note the plus symbol to the left of the method declaration, this is the folding icon and when clicked will fold the method block up, this is a very convenient option which allows the Developer to 'fold' up the code to help readability ...

```
ex1_1.java ✕  
package exercise_1;  
  
public class ex1_1  
{  
    + public static void main(String[] args) {}  
}
```



Using the Java Editor – Folding

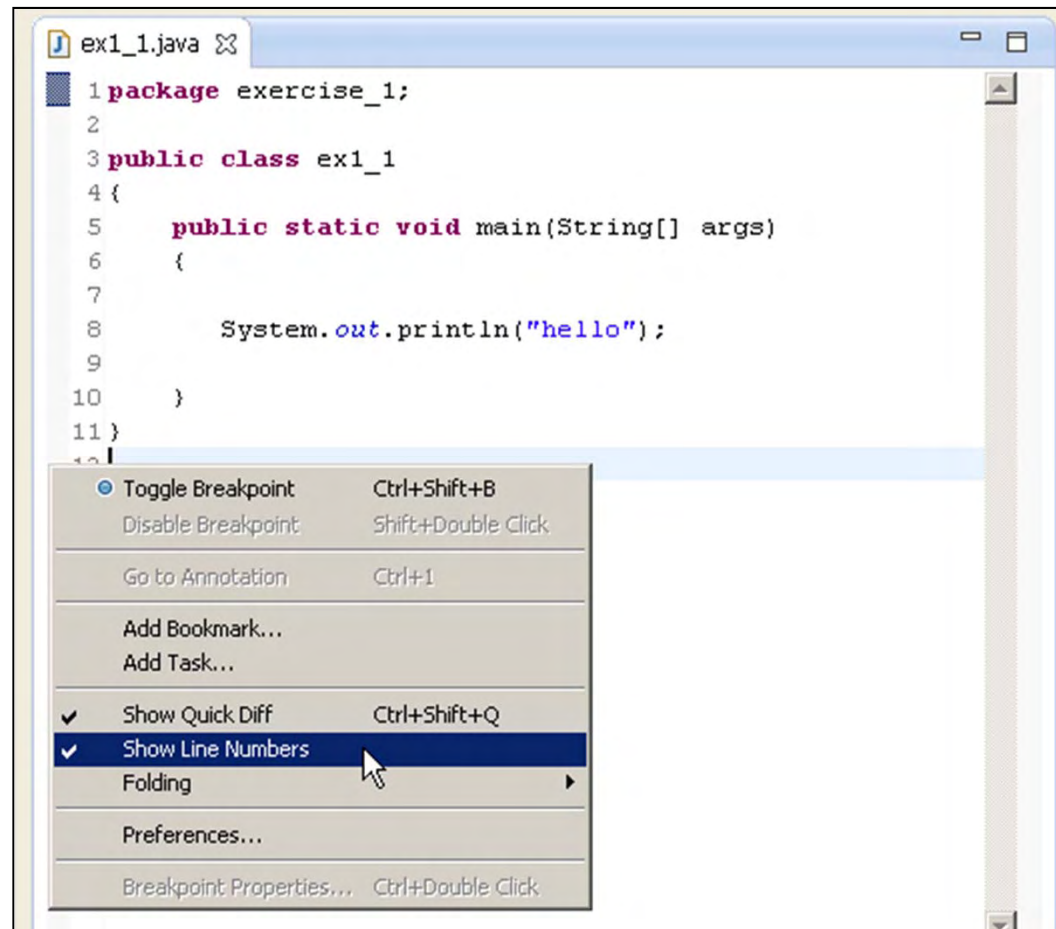
If required the folding icon can be removed from, or added to, the Editor, the Developer will need to right click in the left hand margin





Using the Java Editor – Line Numbers

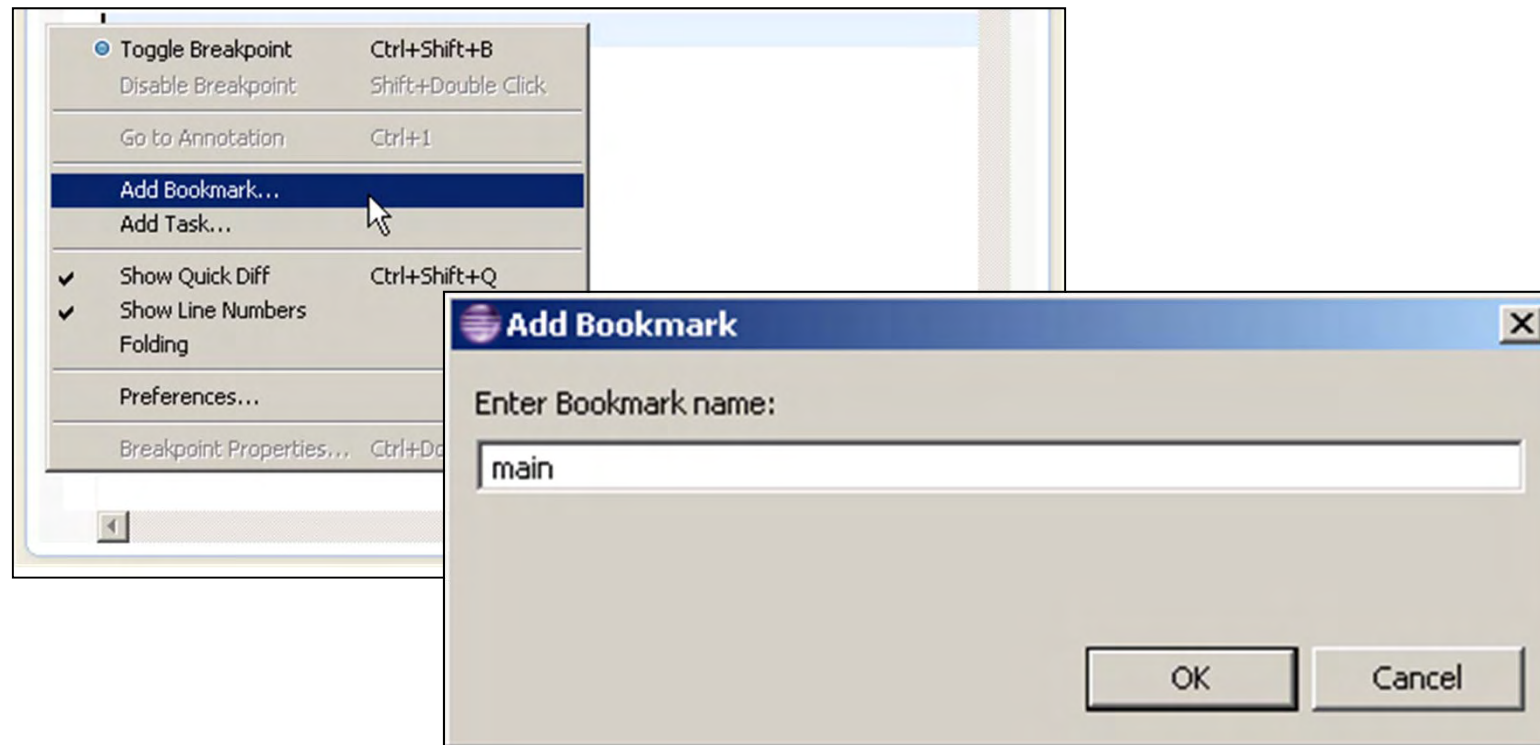
Line numbers in the Editor are very useful when identifying runtime errors and can be switched on/off using the same right click menu ...





Using the Java Editor – Creating Bookmarks

Bookmarks in the Editor are very useful as 'goto' items in the code, they can be created using the same right click menu and allow the Developer to enter a useful name for clarity, note that the bookmark will be placed on the line where the menu is invoked ...



Using the Java Editor – Using Bookmarks



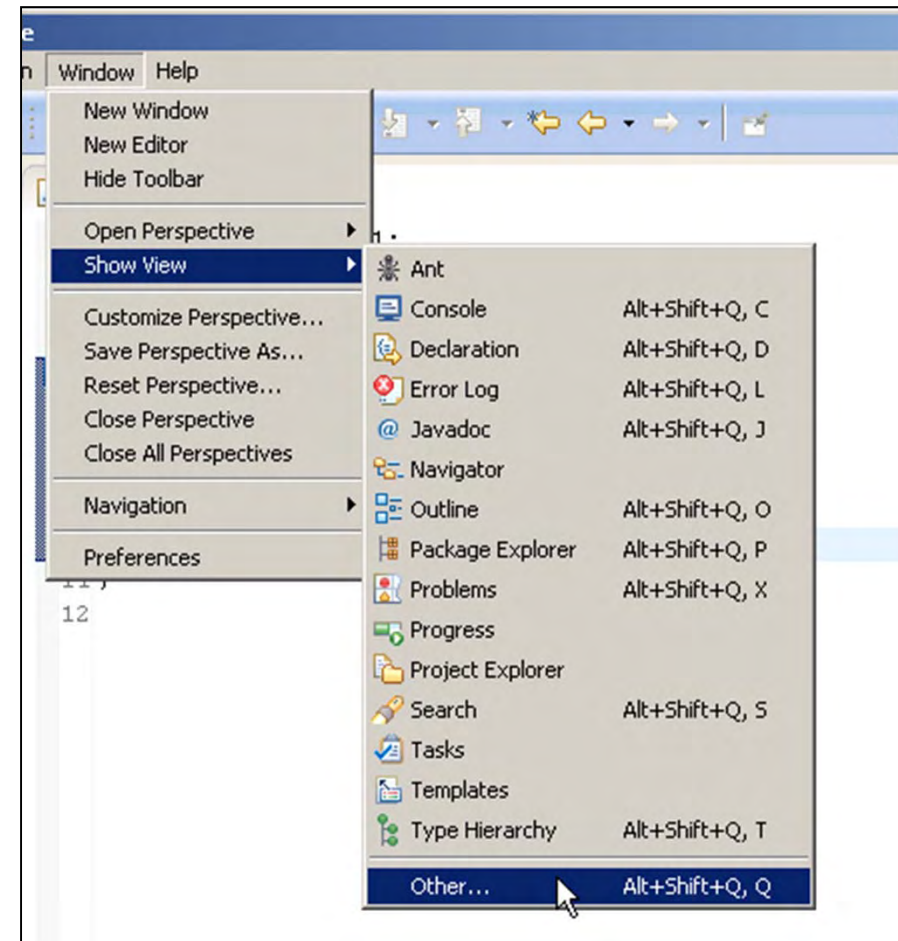
When a Bookmark is created a suitable icon (blue strip) is placed in the left hand margin of the Editor ... when the icon is passed over a tooltip of the bookmark's name will be displayed ...

```
3 public class ex1_1
4 {
5 public static void main(String[] args)
6 {
7
8     System.out.println("hello");
9
10 }
11 }
12
```

Using the Java Editor – Using Bookmarks



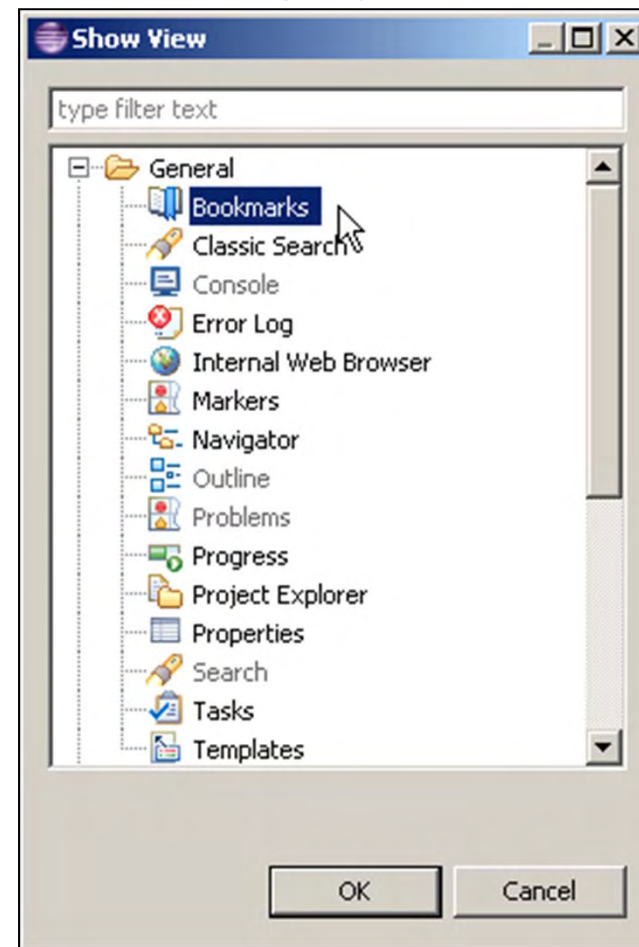
A file's bookmarks can also be displayed in a view in the windows displayed at the bottom (where the console appears), to switch this window on the Developer will need to go to *Windows -> Show View -> Other ...*



Using the Java Editor – Using Bookmarks



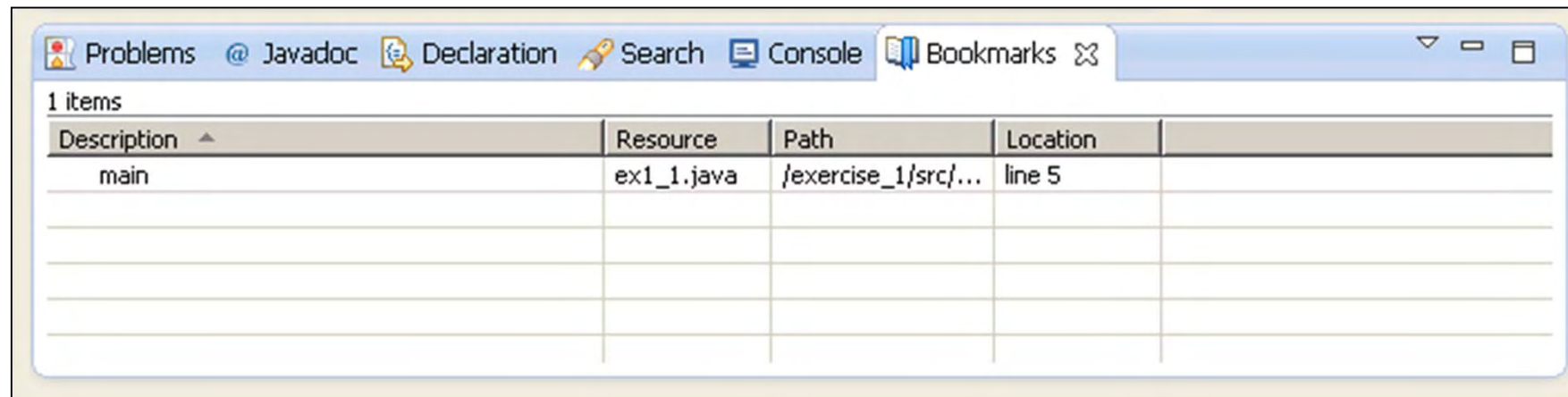
The Developer should then select the Bookmarks view, note that Console, Outline, Problems and Search are greyed out indicating they are already selected ...



Using the Java Editor – Using Bookmarks



The window for the Bookmarks will appear at the bottom and will display all Bookmarks created for the file ...



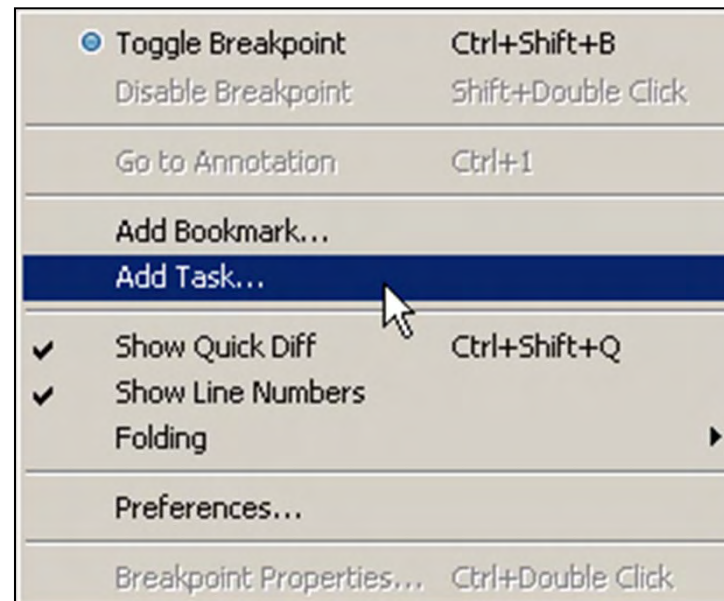
These can be clicked on and the cursor in the Editor will go directly to that Bookmark / line

Using the Java Editor – Using Tasks



Tasks in Eclipse allow the Developer to create a 'to do' list in a similar style to Bookmarks, in fact they are virtually interchangeable except that tasks have an added option of marking them complete as well as setting priorities.

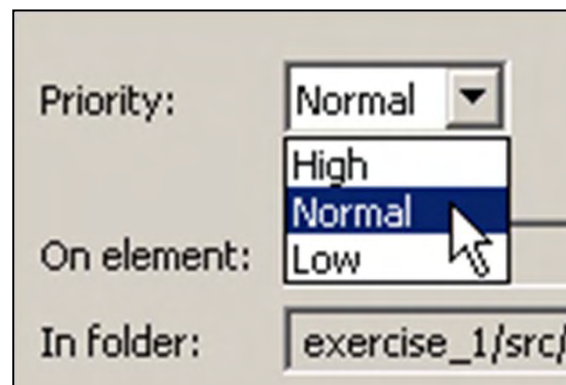
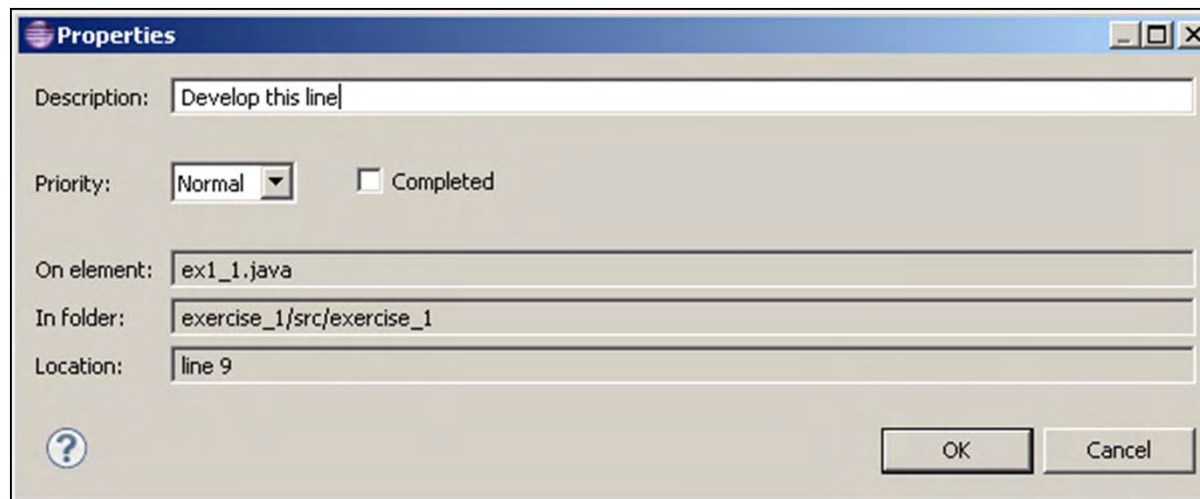
To create a task right click in the margin of the Java Editor



Using the Java Editor – Using Tasks



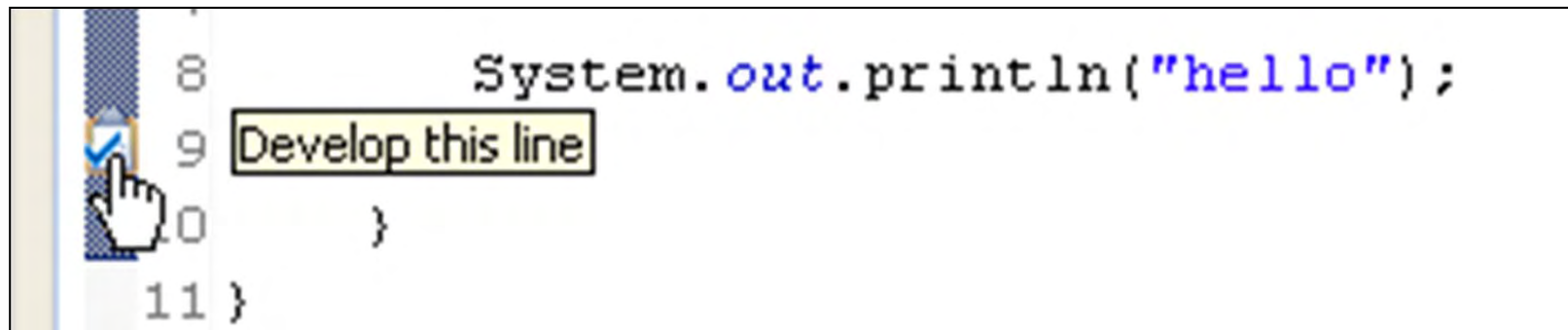
Here the Developer is creating a new task, note the ability to create priority levels ...



Using the Java Editor – Using Tasks

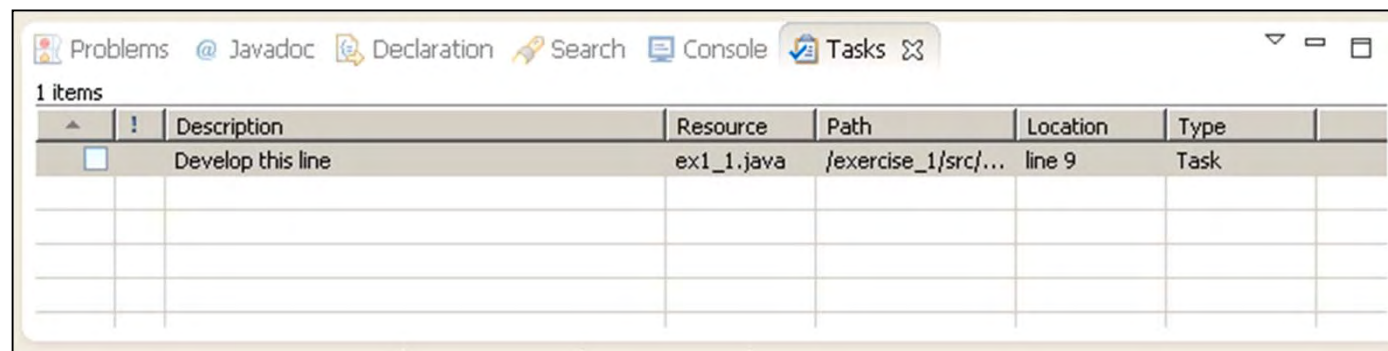
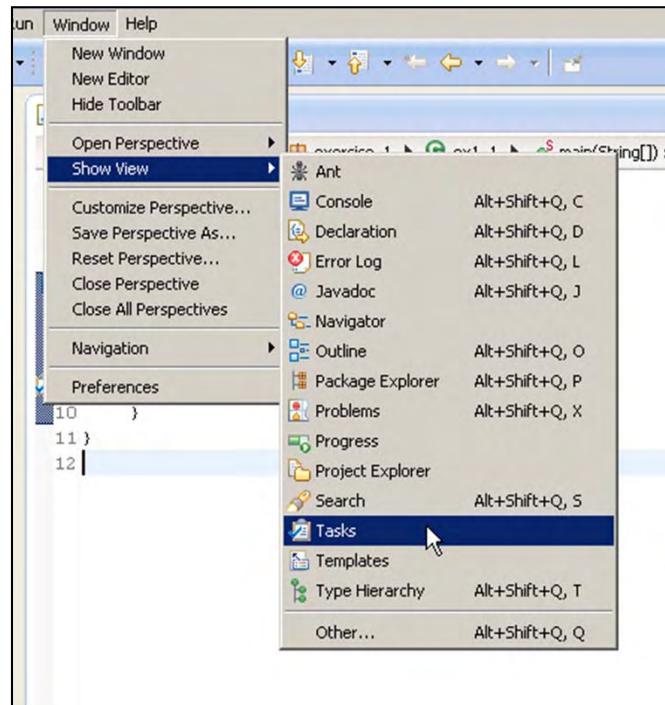


Once created a Task icon will appear in the left hand margin represented by a clipboard with a tick in it. To view the task the Developer can run the mouse over the icon to see the hint ...



The complete list of tasks for the file can also be displayed at the bottom of the software, this can be viewed by selecting Window -> Show View -> Tasks

Using the Java Editor – Using Tasks



Using the Java Editor – Using Tasks



Notice the small box next to the Description, this allows the Developer to mark the Task as completed, this is a very useful feature when using Tasks as 'to do' lists ...

▲	!	Description
<input checked="" type="checkbox"/>		Develop this line

Using the Java Editor – Correcting Errors



One of the great advantages to using the Java Editor is its comprehensive error and syntax checking, during this course we will introduce more aspects of this functionality but will concentrate on simple errors here and examine how the Data Editor deals with them.

Firstly a discussion of what constitutes a syntax error in Java, generally the syntax errors can be coding which will not allow the Java to execute.

The following slide lists some of the problems which can occur ...

Using the Java Editor – Syntax Rules



These rules apply to Java syntax regardless of the size of the program ...

All opened brackets/braces need a matching closing bracket / brace.
All commands / calls in the Program will need to be terminated by a semi-colon.

No semi-colon is necessary following a closing curly bracket (brace).

All strings to be enclosed by double quotes "text"

Numbers are not enclosed by quotes

A line of code can be commented out using //

Lines of code can be commented out using /* and ending with */

Further syntax rules will be introduced throughout the course ...

Using the Java Editor – Error Checking



If the Editor detects an error it will place a red cross on the line in error ...

```
7  
8 System.out.println("hello");  
9  
10 }
```

Passing the mouse over the cross will display the perceived error ...

```
7  
8 Syntax error, insert ";" to complete BlockStatements  
9  
10 }
```

Using the Java Editor – Error Checking



Adding a semi-colon will terminate the command and clear the red cross.

If the Developer chooses to ignore the error and attempt to run the code, the error will appear in the Console area and the program will not run ...

A screenshot of the Eclipse IDE's Console window. The window title bar includes 'Problems', 'Javadoc', 'Declaration', 'Search', 'Console', and 'Tasks'. The console output shows a terminated Java application with the following error message:

```
<terminated> ex1_1 (1) [Java Application] C:\Program Files\Java\jdk1.6.0_33\bin\javaw.exe (29 Aug 2013 15:21:21)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Syntax error, insert ";" to complete BlockStatements

    at exercise_1.ex1_1.main(ex1_1.java:8)
```

Using the Java Editor – Colour Coding



The Java Editor will highlight commented out code in a light blue colour ...

```
16  /**
17   * @param args
18   * @author Tom
19   * @throws ArithmeticException
20   */
21  public static void main (int[] args) throws ArithmeticException
22  {
23
24      try {
25  |   char sex = 'F';
26
27
28          if (sex != 'F')
29
30              throw new RuntimeException("Sex value is incorrect");
31
```

Blue is used for Strings and characters, while reserved words are displayed in bold.