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 …

Using the Java Editor – Colour Coding

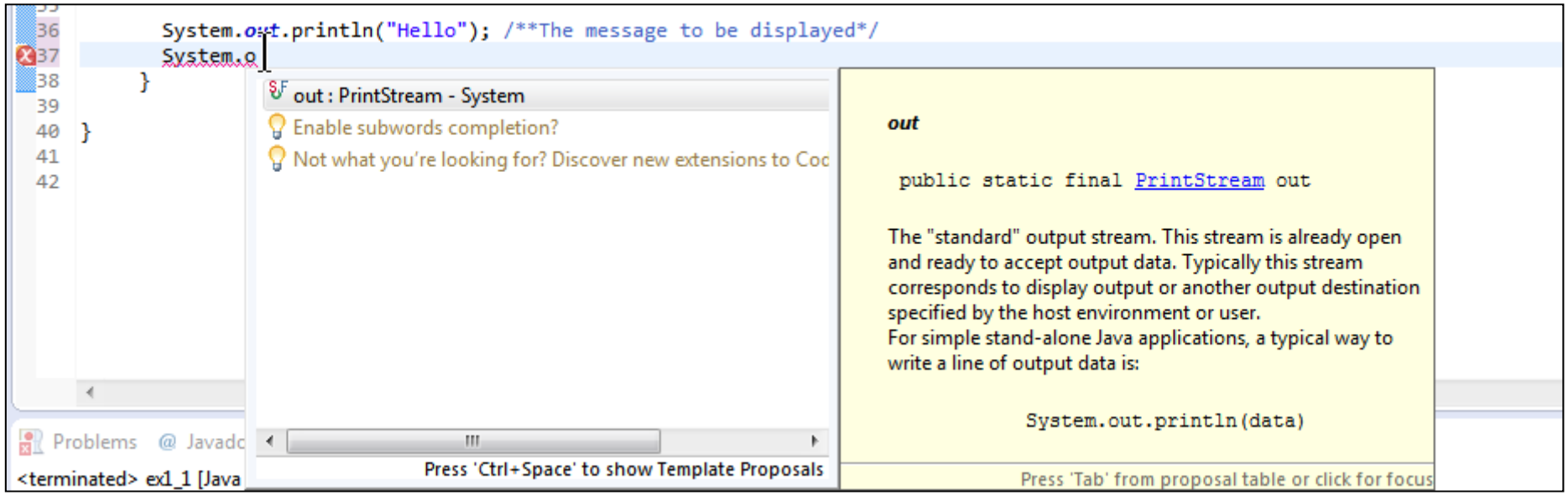
The Java Editor will highlight commented out code in a light blue colour (when using /**, single line comments are green) …

```
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
```

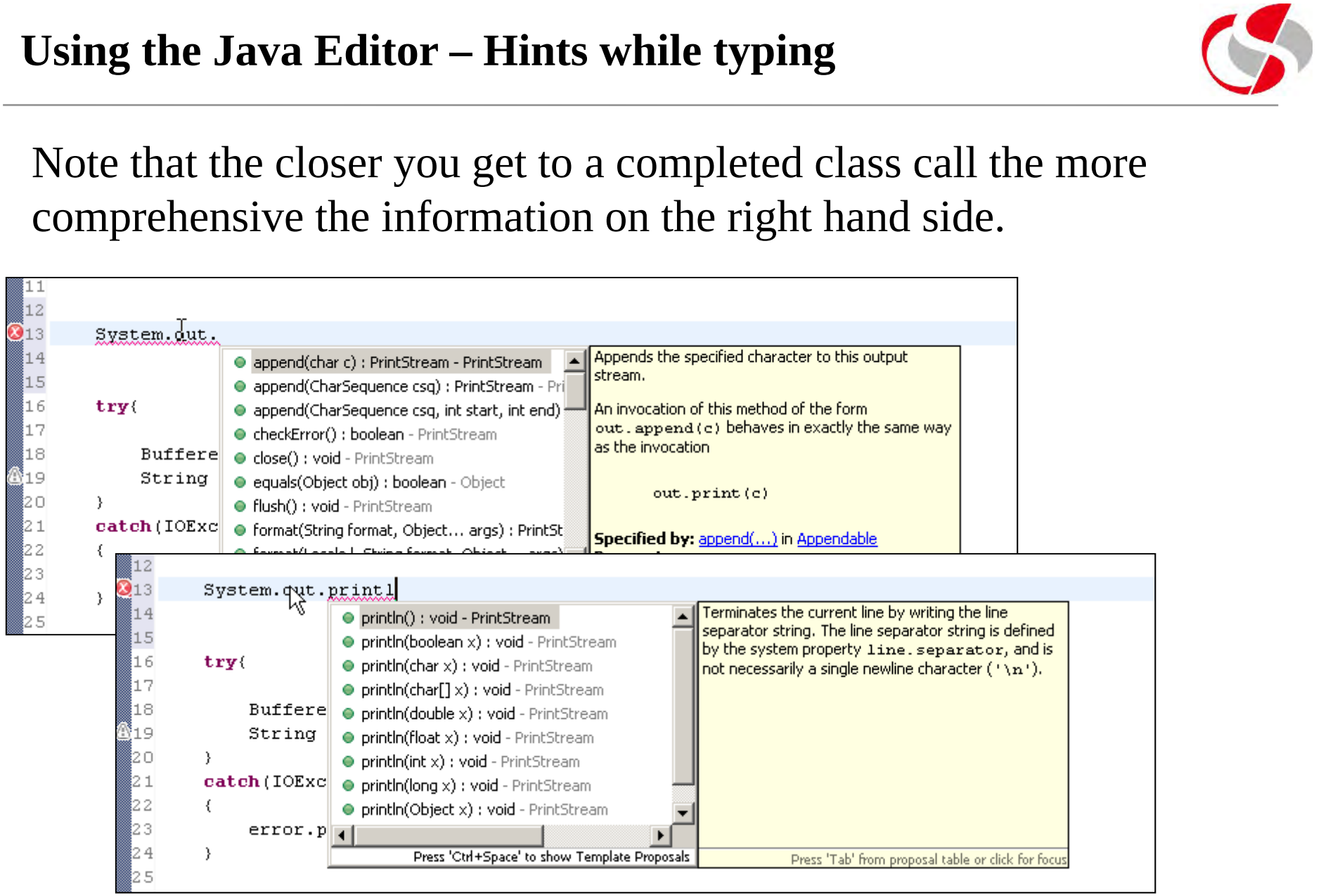
A dark blue is used for Strings and characters, while reserved words are displayed in black.

Using the Java Editor – Hints while typing

The Java Editor will detect familiar patterns when the Developer is typing and will open up a window with hints for completion, it will also report the line as an error until complete !! …



Using the Java Editor – Hints while typing



Note that the closer you get to a completed class call the more comprehensive the information on the right hand side.

Using the Java Editor – Bracket Location

Java code is very dependant on brackets, braces and parentheses being matched, that is, an opened brace must have a closing brace, the editor in Eclipse will match the brace when clicked on …

```
public static void main(String[] arguments)
{

   System.out.println("Hello");

}
```

```
public static void main(String[] arguments)
{

   System.out.println("Hello");

}
```