

# **3 Getting to know your Programming Environment**



# Objectives

At the end of the lesson, the student should be able to:

- Create a Java program using text editor and console in the Linux, Solaris, Windows, Mac OS, or any other OS environment
- Differentiate between syntax-errors and runtime errors
- Create a Java program using NetBeans



# Definitions

- Console
  - This is where you type in commands
  - Examples are Terminal (Linux), MSDOS Command Prompt (Windows)



# Definitions

- Text Editor
  - Examples: Notepad, Wordpad, Vi



# Definitions

- Integrated Development Environment or IDE
  - a programming environment integrated into a software application that provides a GUI builder, a text or code editor, a compiler and/or interpreter and a debugger.



# My First Java Program

```
1 public class Hello {  
2  
3     /**  
4      * My first Java program  
5      */  
6     public static void main( String[] args ){  
7  
8         //prints the string "Hello world" on screen  
9         System.out.println("Hello world");  
10  
11     }  
12 }
```



# Using Text Editor and Console

- NOTE:
  - This will be demonstrated by the teacher
  - Environment used: Red Hat Linux
  - For Windows Environment: Refer to Appendix B in your student Manual



# Using Text Editor and Console

- Step 1: Start the Text Editor
  - To start the Text Editor in Linux, click on Menu-> Accessories-> Text Editor
- Step 2: Open Terminal
  - To open Terminal in Linux, click on Menu-> System Tools-> Terminal
- Step 3: Write your the source code of your Java program in the Text Editor





# Using Text Editor and Console

- Step 4: Save your Java Program
  - Filename: Hello.java
  - Folder name: myjavaprograms
  - To open the Save dialog box, click on the File menu found on the menubar and then click on Save.
  - If the folder myjavaprograms does not exist yet, create the folder



# Using Text Editor and Console

- Step 5: Compiling your program
  - Go to the Terminal window
  - Go to the folder myjavaprograms where you saved the program
  - To compile a Java program, we type in the command:  
`javac [filename]`
  - So in this case, type in:  
`javac Hello.java`

**During compilation, javac adds a file to the disk called [filename].class, or in this case, Hello.class, which is the actual bytecode.**



# Using Text Editor and Console

- Step 6: Running the Program
  - To run your Java program, type in the command:  
`java [filename without the extension]`
  - so in the case of our example, type in:  
`java Hello`
  - You can see on the screen after running the program:  
`"Hello world!"`

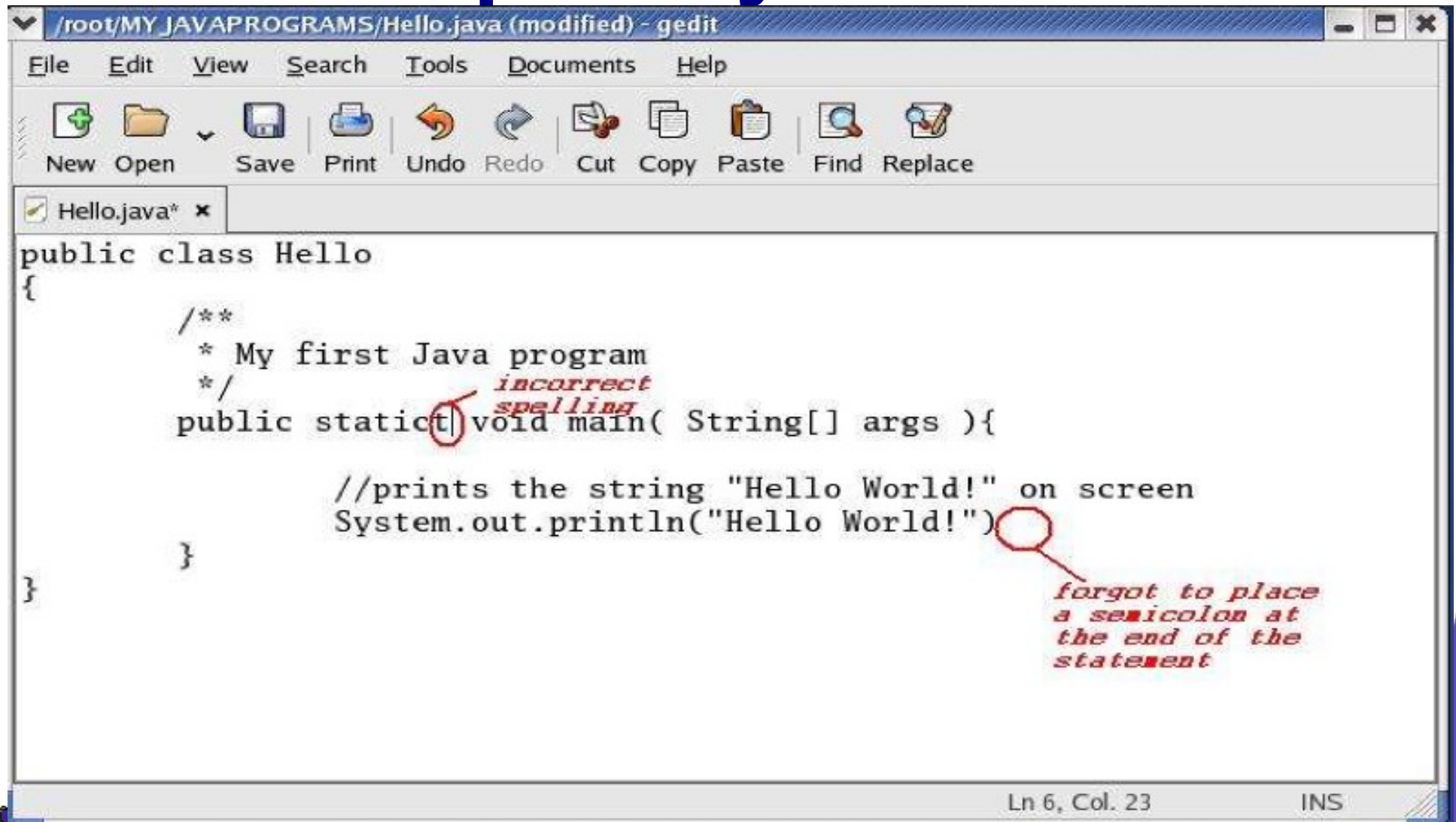


# Errors : Syntax Errors

- Syntax Errors Syntax
  - errors are usually typing errors
- Common Syntax Errors:
  - misspelled a command in Java
  - forgot to write a semi-colon at the end of a statement



# Example: Syntax Error



The screenshot shows a gedit editor window titled "/root/MY\_JAVAPROGRAMS/Hello.java (modified) - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for New, Open, Save, Print, Undo, Redo, Cut, Copy, Paste, Find, and Replace. The editor window displays the following Java code:

```
public class Hello
{
    /**
     * My first Java program
     */
    public static void main( String[] args ){
        //prints the string "Hello World!" on screen
        System.out.println("Hello World!")
    }
}
```

Two red annotations highlight syntax errors:

- A red circle around the word "void" in the `public static void main` line, with the text *incorrect spelling* written next to it.
- A red circle around the closing parenthesis of the `System.out.println` statement, with a line pointing to it and the text *forgot to place a semicolon at the end of the statement*.

The status bar at the bottom right shows "Ln 6, Col. 23" and "INS".



# Example: Syntax Error

```
root@localhost:~/MYJAVAPROGRAMS
File Edit View Terminal Go Help
[root@localhost MYJAVAPROGRAMS]# javac Hello.java
Hello.java:6: <identifier> expected
    public statict void main( String[] args ){
            ^
Hello.java:11: ';' expected
}
^
2 errors
[root@localhost MYJAVAPROGRAMS]#
```



# Errors: Runtime Errors

- Run-time Errors
  - errors that will not display until you run or execute your program
  - Even programs that compile successfully may display wrong answers if the programmer has not thought through the logical processes and structures of the program.
  - Examples:
    - You want your program to print 100 strings of “Hello world”, but it only printed 99.
    - Your program gets an input from the user, but the user inputted a character, and so your program crashes/terminates



# Summary

- My First Java Program
- Using a Text Editor and Console
  - Write program
  - Compile program
  - Run program
- Errors
  - Syntax Errors
  - Runtime Errors

