

# Creating a Java Main Class



# Objectives

After completing this lesson, you should be able to:

- Use the NetBeans IDE to create and test Java classes
- Write a main method
- Use System.out.println to write a String literal to system output



# Topics

- Java classes and packages
- The `main` method



# Java Classes

A Java class is the building block of a Java application.

ShoppingCart.java

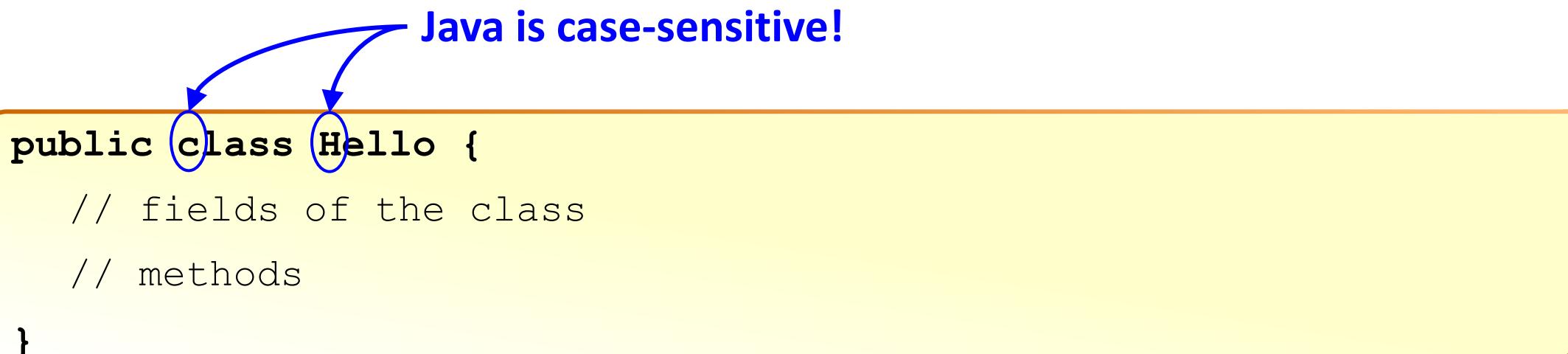
←Includes code that:

- Allows a customer to add items to the shopping cart
- Provides visual confirmation to the customer

# Program Structure

- A class consists of:
  - The class name. Class names begin with a capital letter.
  - The body of the class surrounded with braces { }
  - Data (called fields)
  - Operations (called methods)
- Example:

**Java is case-sensitive!**



```
public class Hello {  
    // fields of the class  
    // methods  
}
```

# Java Packages

- A package provides a namespace for the class.
  - This is a folder in which the class will be saved.
  - The folder name (the package) is used to uniquely identify the class.
  - Package names begin with a lowercase letter.
- Example:

package greeting;

```
public class Hello {  
    // fields and methods here  
}
```

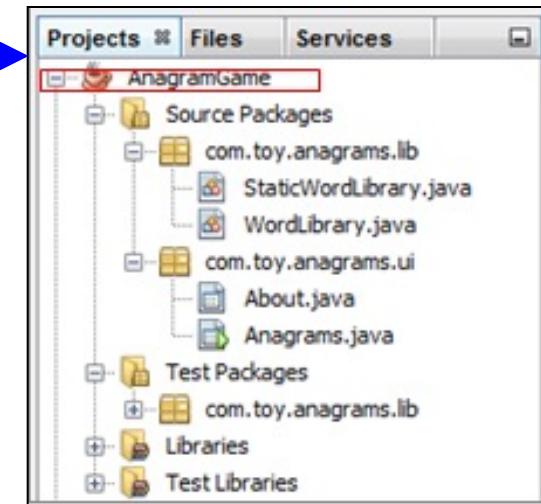
Package name

The class's unique name is:  
greeting.Hello

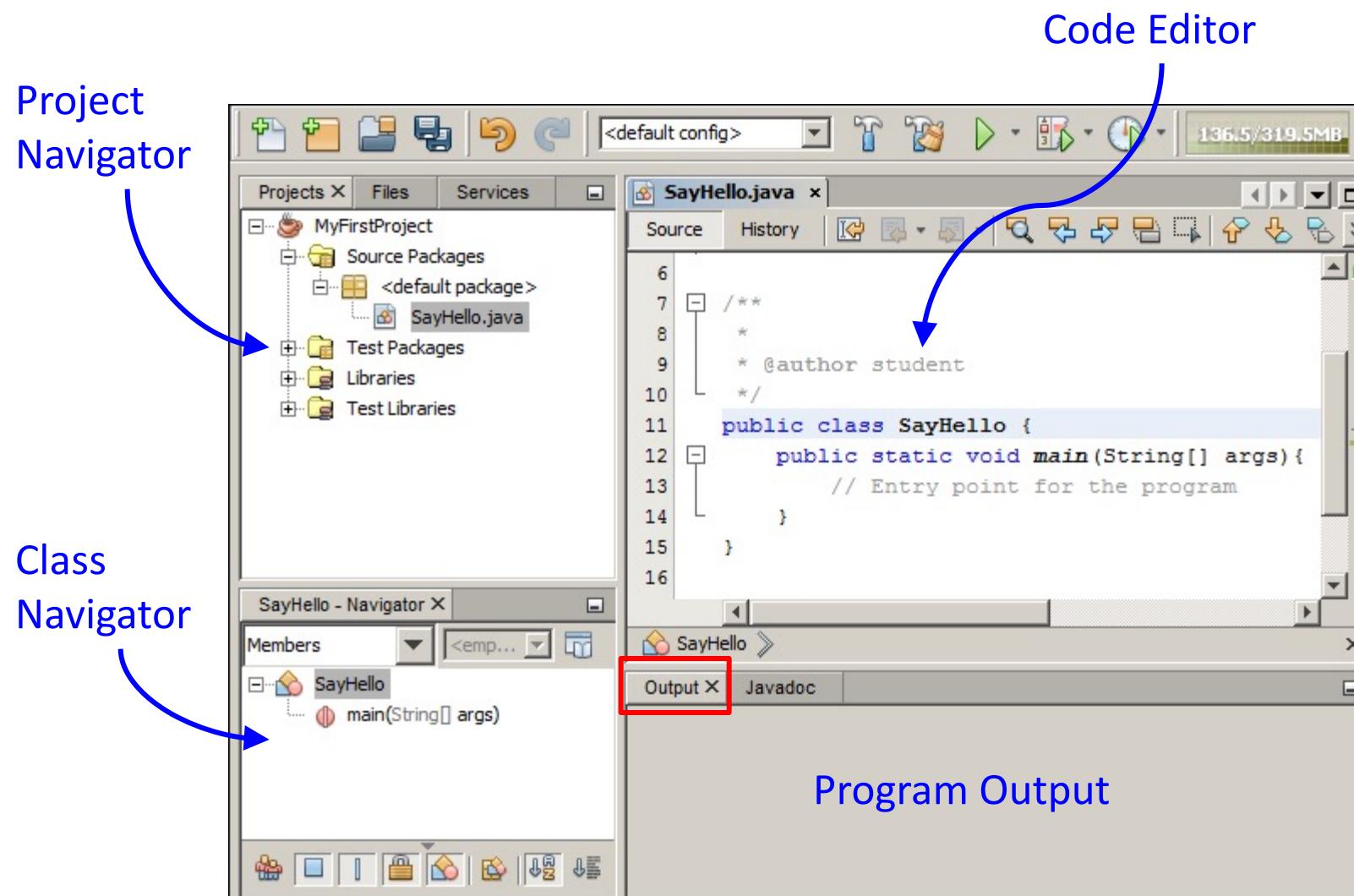
# Java IDEs

A Java Integrated Development Environment (IDE) is a type of software that makes it easier to develop Java applications.

- An IDE provides:
  - Syntax checking
  - Various automation features
  - Runtime environment for testing
- It enables you to organize all your Java resources and environment settings into a *Project*. 
- Projects contain packages.
- Packages contain files, such as .java.

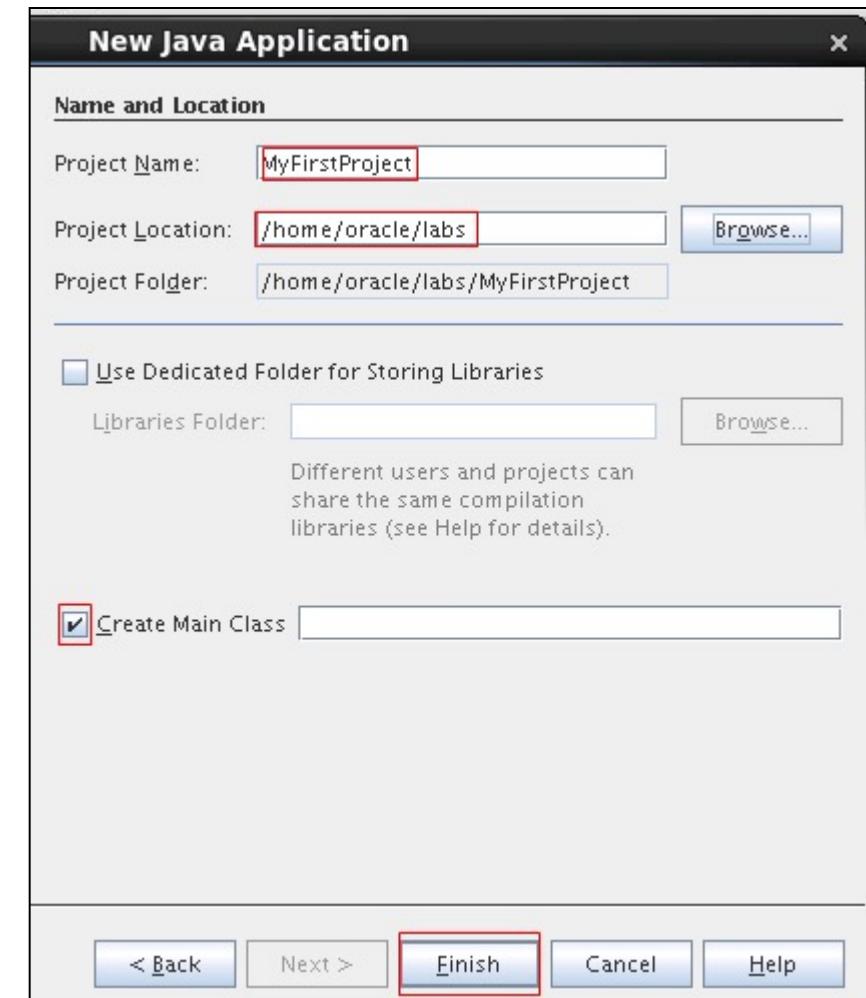


# The NetBeans IDE



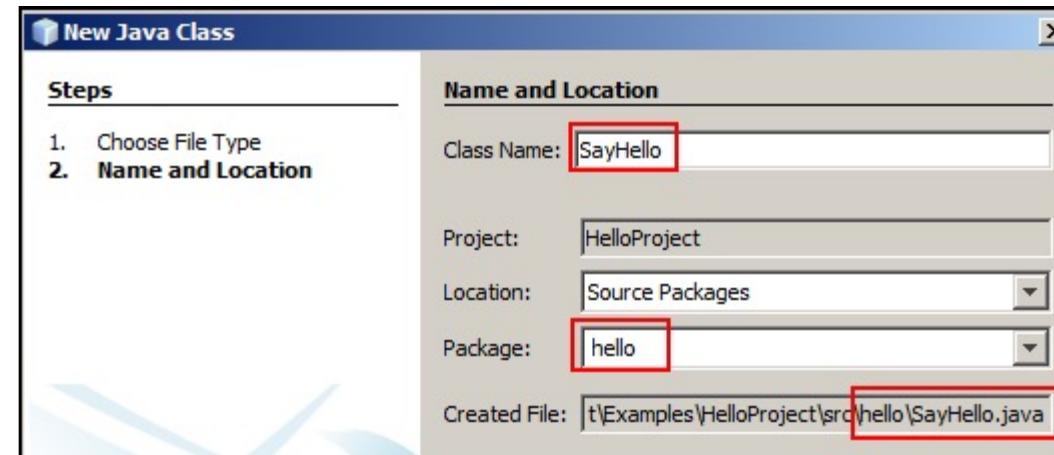
# Creating a Java Project

1. Select **File > New Project**.
2. Select Java Application.
3. Name and set the location for the project.
4. Select “Create Main Class” if you want it done for you automatically.
5. Click **Finish**.



# Creating a Java Class

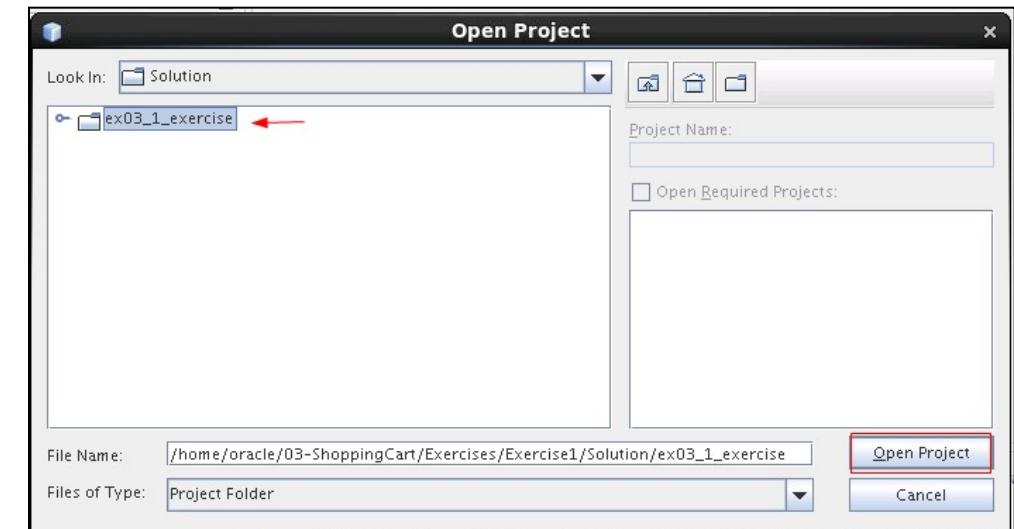
1. Select **File > New File**.
2. Select your project and choose **Java Class**.
3. Name the class.
4. Assign a package.
5. Click **Finish**.



# Opening an Existing Java Project

If you need to open an existing project in NetBeans, perform the following steps:

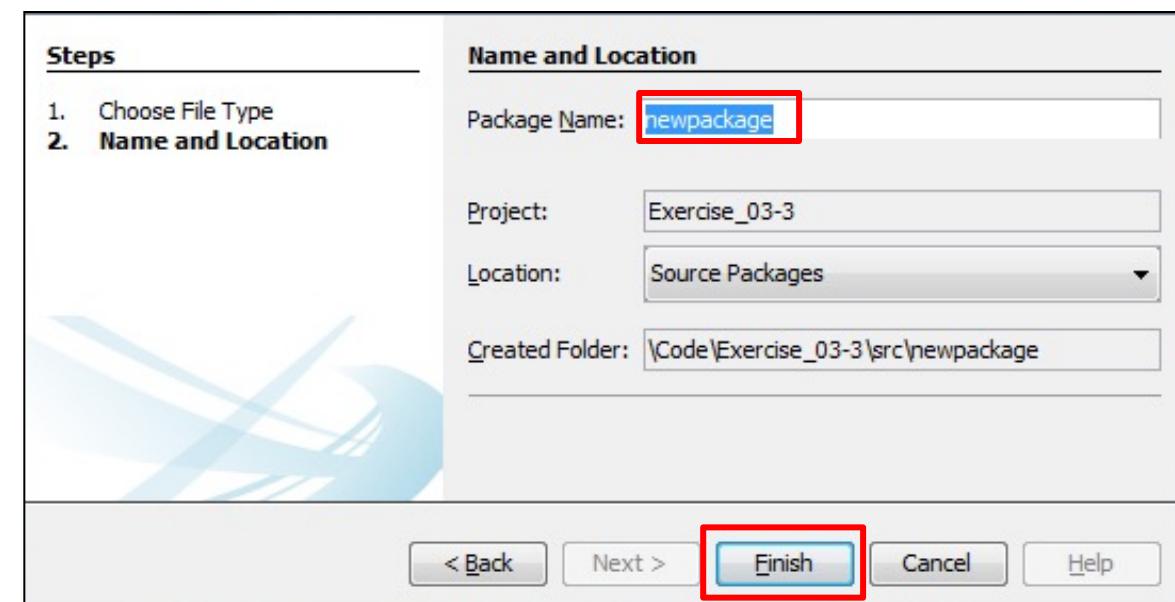
1. Select **File > Open Project**.
2. Navigate to the directory that contains your projects.
3. Select the project file you want. (This file must be unzipped.)
4. Click **Open Project**.



# Creating a New Java Package

If you ever need to create a new package, perform the following steps in NetBeans:

1. Right-click your project.
2. Select **New > Java Package**.
3. Name the package.
4. Click **Finish**.



# Topics

- Java classes and packages
- The `main` method



# The main Method

- It is a special method that the JVM recognizes as the starting point for every Java program.
- The syntax is always the same:

```
public static void main (String[] args) {  
    // code goes here in the code block  
}
```

- It surrounds entire method body with braces { } .

# A main Class Example

```
public class Hello {  
  
    public static void main (String[] args) {  
        // Entry point to the program.  
        // Write code here:  
        System.out.println ("Hello World!");  
    }  
}
```

Class name

Comments

main  
method

Program output

# Output to the Console

- Syntax:

```
System.out.println (<some string value>);
```

- Example:

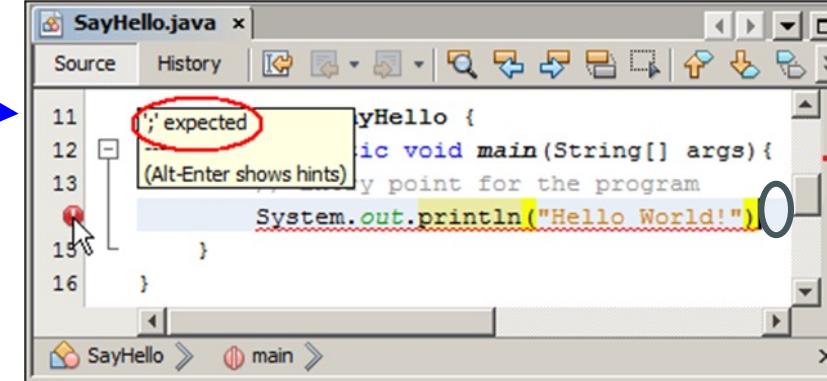
```
System.out.println ("This is my message.");
```

String literal

Be sure to include the  
semicolon

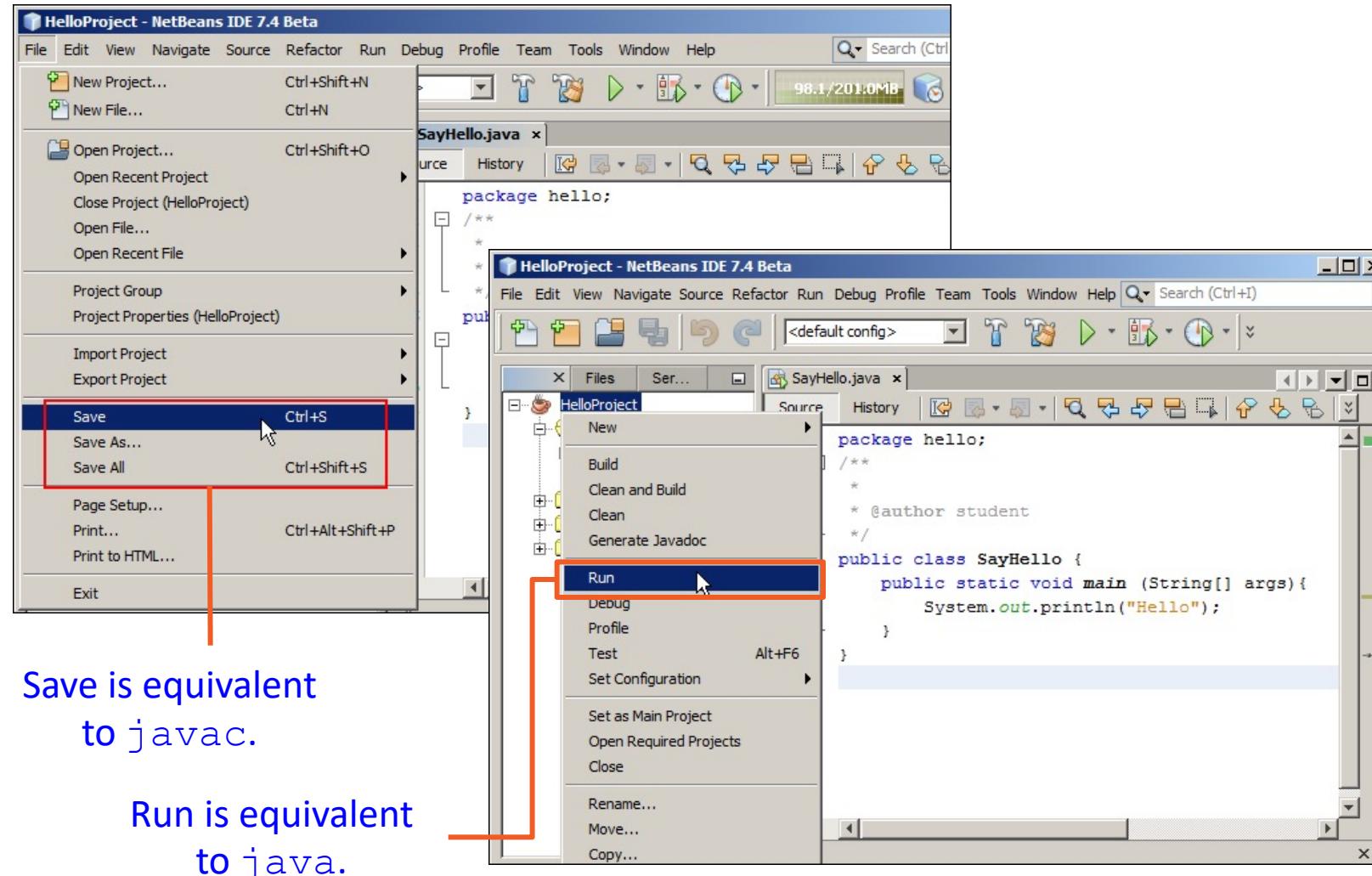
# Avoiding Syntax Errors

- NetBeans will tell you if you have done something wrong.
- Common errors include:
  - Unrecognized word (check for case-sensitivity error)
  - Missing close quotation mark
  - Unmatched brace
  - Missing semicolon



The screenshot shows the NetBeans IDE interface with a Java file named "SayHello.java". The code contains a syntax error: "System.out.println("Hello World!");" is missing a closing brace at the end of the main method. A red circle highlights the word "println" in yellow, and a blue arrow points from the "Missing semicolon" bullet point in the list above to this error. A tooltip "'; expected" is displayed over the opening brace of the println statement, and a status bar at the bottom shows "SayHello > main >".

# Compiling and Running a Program by Using NetBeans



# Quiz



Which main method syntax is correct?

- a. Public static void main (String[ ] args) { }
- b. public Static void Main (String[ ] args) { }
- c. public static void main (String ( ) args) [ ]
- d. public static void main (String[ ] args) { }



# Summary

In this lesson, you should have learned how to:

- Use the NetBeans IDE to create and test Java classes
- Write a `main` method
- Use `System.out.println` to write a String literal to system output

