
ORACLE
ACADEMY

Java Fundamentals

4-1
Getting Started with Eclipse

ORACLE ACADEMY

Objectives

This lesson covers the following objectives:

- Identify components of Eclipse
- Identify components of a Java application
- Compile an application
- Test to ensure application is complete
- Write the code for GalToLit.java

Objectives

This lesson covers the following objectives:

- Modify a program to execute error free
- Modify a program to use a formula to convert units of measure

Eclipse Community and Requirements

- Facts about Eclipse:
  - Eclipse was created by an Open Source community.
  - The Eclipse project is governed by the Eclipse Foundation, a non-profit organization.
  - Eclipse requires an installed Java Runtime Environment (JRE).
  - Eclipse contains its own development tools, i.e., a Java compiler.

Java JRE and Java JDK

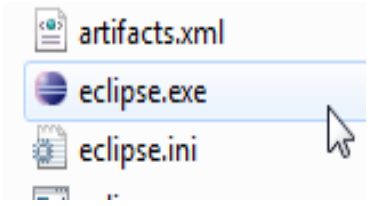
- Differences between Java JRE and Java JDK:
  - Java Runtime Environment (JRE) contains only the necessary functionality to start Java programs, such as Internet applications.
  - Java Development Kit (JDK) contains functionality to start Java programs as well as develop them.
  - At a minimum, the Java JRE is required to run Eclipse.

Check for Java on Your Computer

- To verify that Java is already installed on your computer:
- Windows or Linux operating systems:
  - Enter java -version in a command window.
- Mac operating system:
  - Use the Software Update option from the Apple menu.
- This course assumes that you have Java and Eclipse installed on your computer.

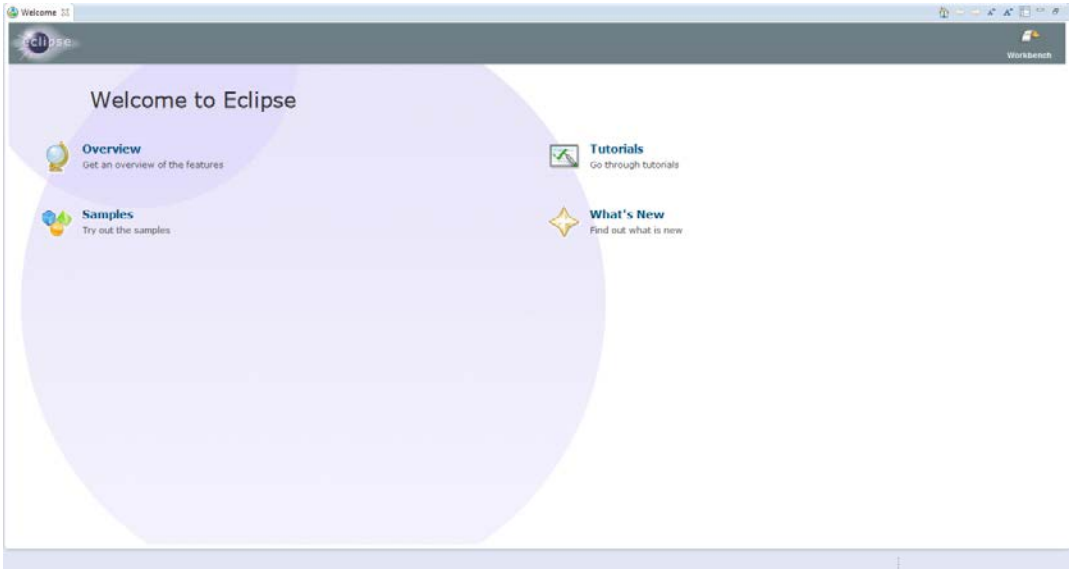
Steps to Launch Eclipse

- On a computer with Windows double-click on the file eclipse.exe. On a Linux or Mac computer double click on the file eclipse.
- When prompted, enter the pathname for the workspace into which you will store your Java projects and click the OK button. This can be your c:\ drive, or possibly a network drive.
- Eclipse will start and display the Welcome page.
- Close the Welcome page by clicking the X next to the Welcome tab name.



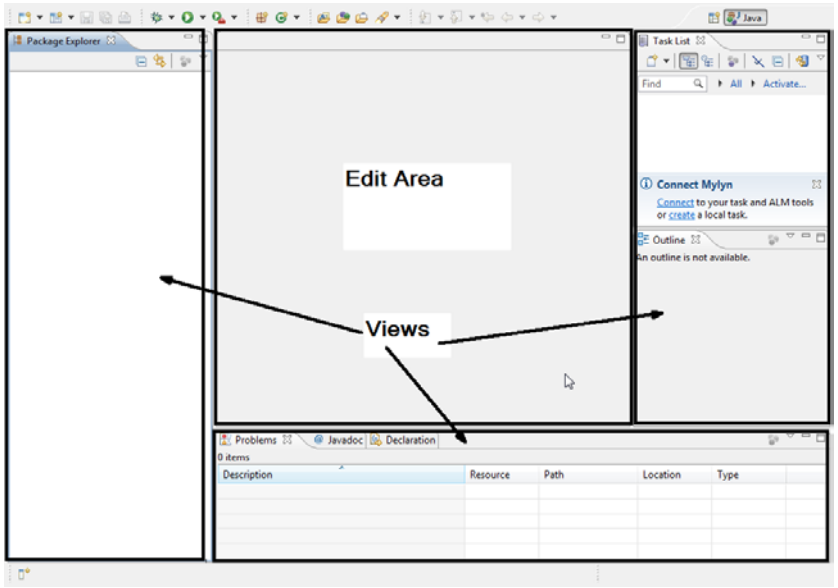
Eclipse Welcome Page

- There are valuable resources available from the Welcome page.
- You can return to the Welcome page by choosing Welcome from the Help menu.



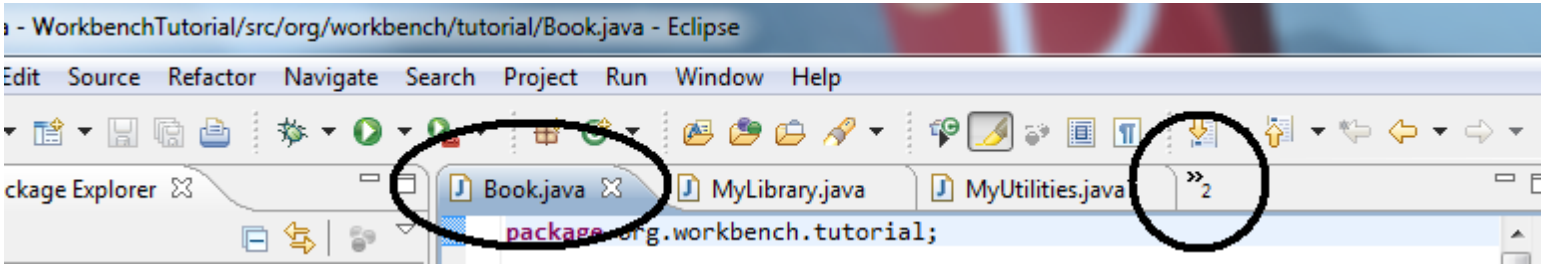
Eclipse Edit Area and Views

- Eclipse provides an edit area and several views.
- An editor is where you type in your Java source code.
- Views are sub-windows that provide information about your project.



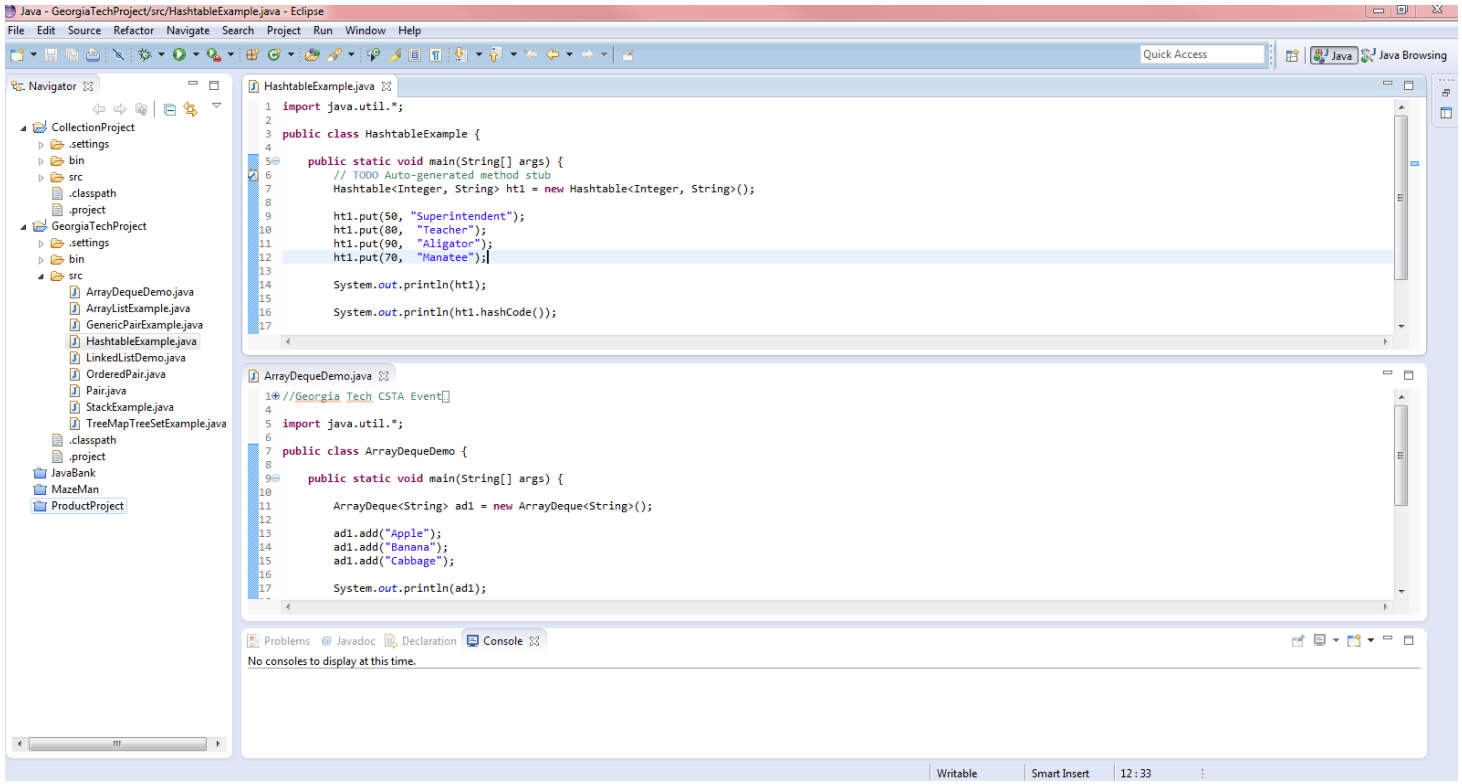
Eclipse Edit Area Tabs

- The edit area uses tabs when more than one file is open.



Eclipse Edit Area Windows

- The edit area can have multiple windows occupy the space.



Additional Details on Edit Areas and Views

- A combination of views and editors are referred to as a perspective.
- You can choose Open Perspective from the Window menu.

The Workspace

- All projects are developed and modified in a workspace.
- A workspace is a collection of Projects.
- In this course, you may use the same workspace for all practice projects and packages.
- A project is a way for programmers to organize Java files.
- A package is how Java and Eclipse organize Java files that are related.
- Using packages will ensure that related files can find each other.

Switching Workspaces

- You can Switch Workspaces (from the File menu) to change to a different physical location for your files.

High-Level Steps to Create a Program in Eclipse

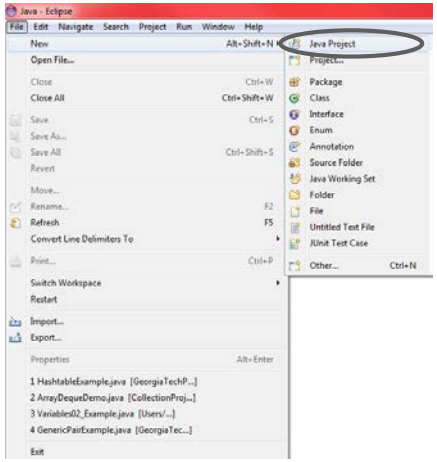
- Create a Project.
- Create a Package (inside the src folder of the project).
- Create Class(es) inside the package.
  - At least one of the classes must contain a main method. This class is called the Driver class.
- Compile the Java code. This creates a .class file.
- Run the Java code from the Driver class.

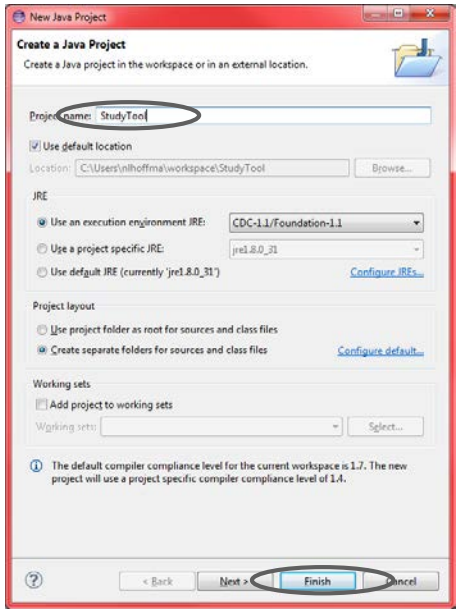
Projects in Eclipse

- In Eclipse:
  - All programs must reside inside a project for proper compilation.
  - You may have one or multiple class files in one project.
  - One of the classes must contain a main method.

Steps to Create a Project in Eclipse

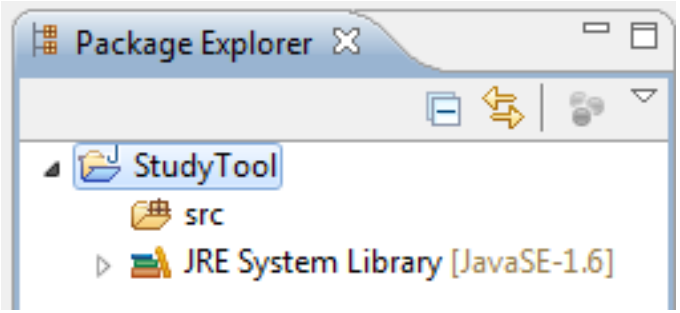
- Choose File → New → Java Project*.
- Enter a project name and click Finish.
- All project values are set to default by clicking the Finish button.





Project Display

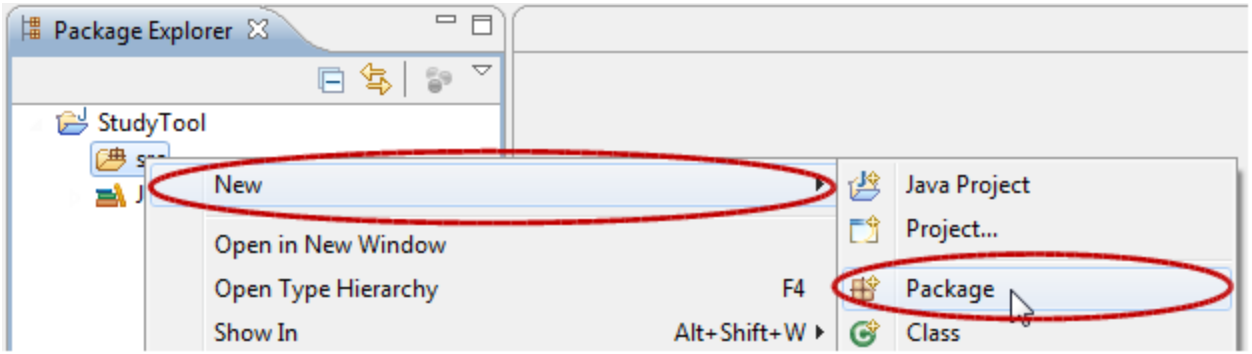
- The project is created and displayed as a folder.
- It displays in the Package view to the left of the edit area.



Steps to Create a Package

- Select the src folder in the Package view.
- Right click the src folder and choose New → Package.

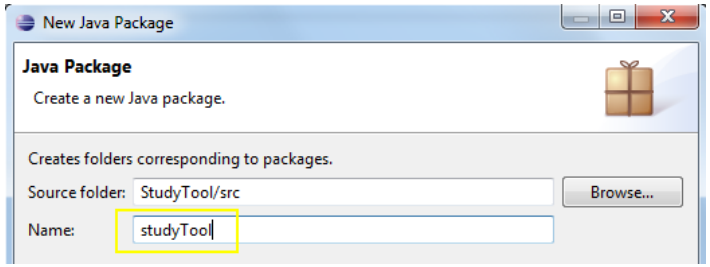
A package, in Java, is a mechanism for organizing Java classes into namespaces or containers. In Eclipse, packages are created inside projects.

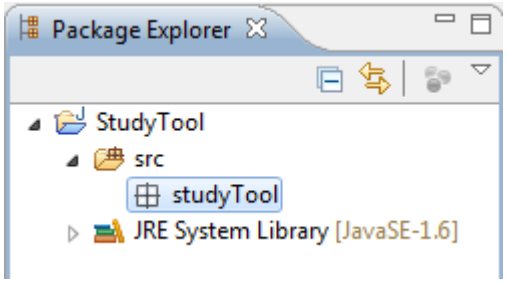


Naming Packages

- Name this package the same name as the Project using lower camel case.

Camel case is the practice of stringingCapitalizedWords together with no spaces. Lower camel case does not capitalize the lead word.

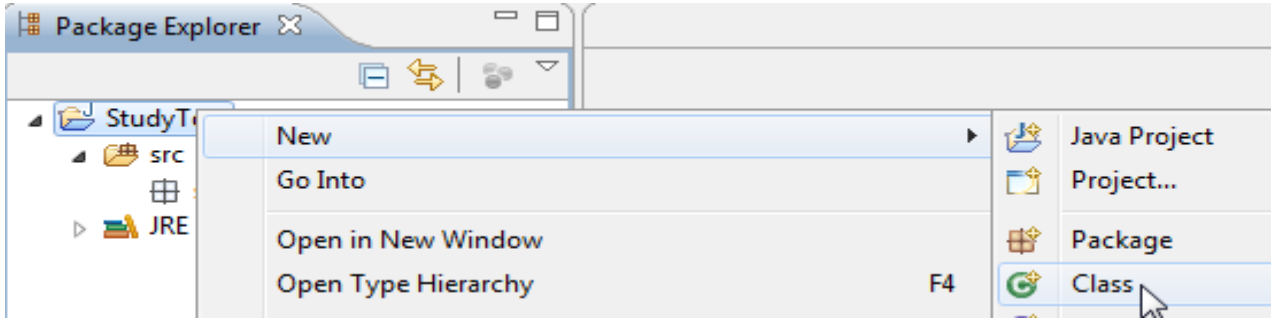




Steps to Create a Class

1. Right click on the project name to create a new class named StudyPage in the studyTool package.
2. Select the option to create the main method.

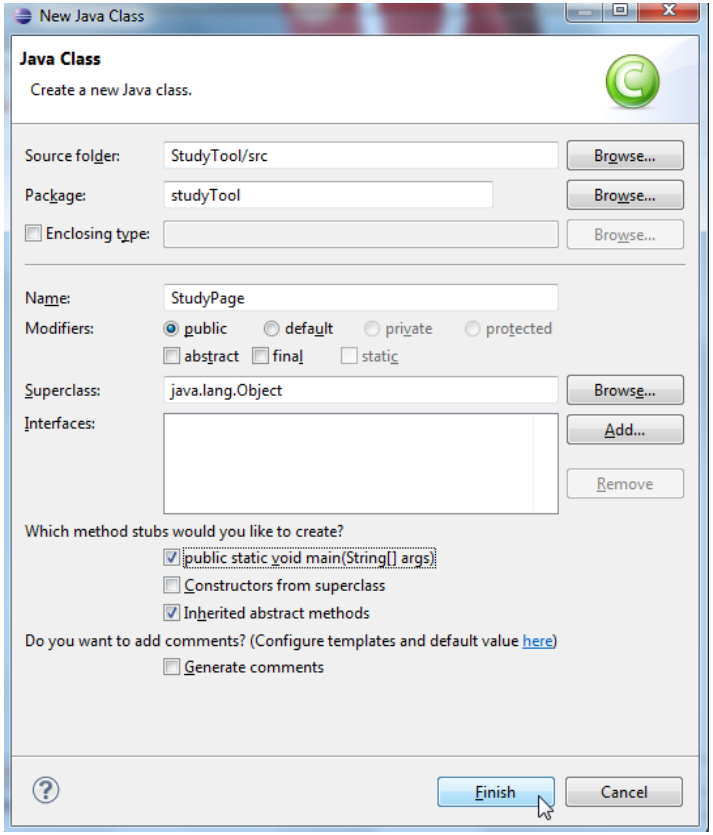
A class in Java is a construct that is used as a blueprint to create objects. It can also be a construct in which objects are created.



Create a Class Display

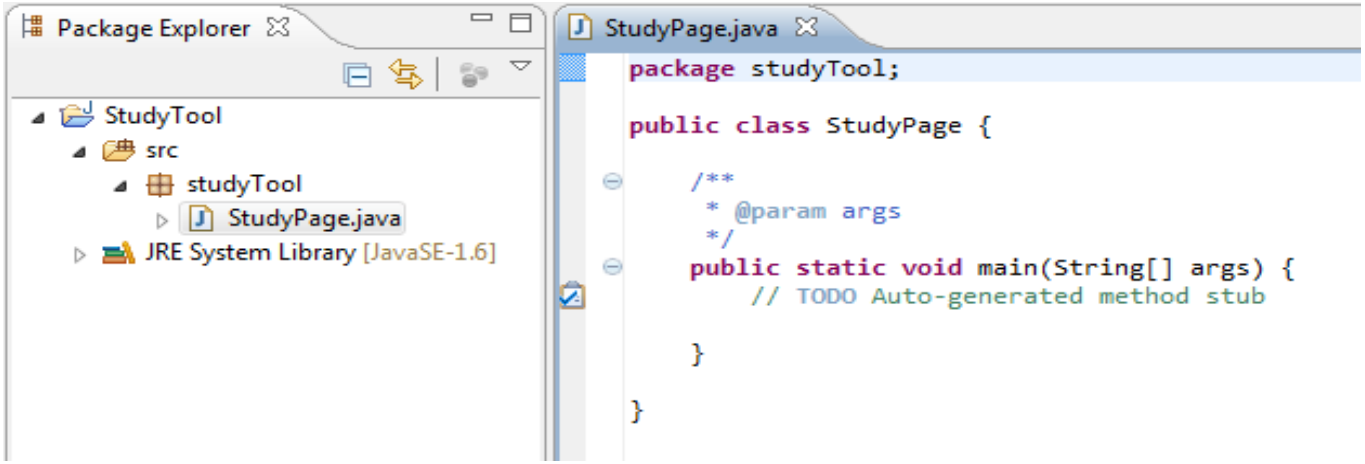
- The option is selected to create the main method.

A main method, in Java, is the method inside a class that runs when the class is compiled and run. This class is referred to as the Driver Class.



New Class

- Notice the following:
  - The StudyPage.java class appears in the studyTool package in the Package Explorer View.
  - The StudyPage.java class is created with the main method.



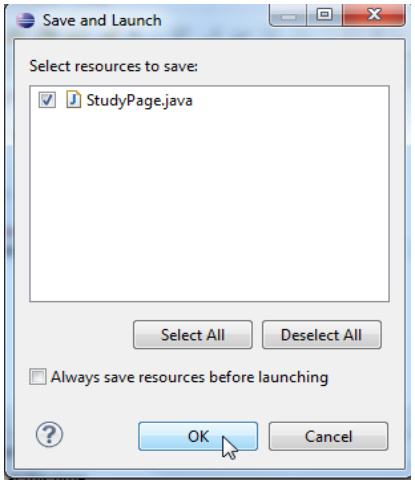
Steps to Create and Run Java Code

- To implement the main method and run the program, enter the following information into the main class.
- The method inside a class that runs when the class is compiled and ran is the main method.

```
public class StudyPage {
  public static void main(String[] args) {
    System.out.println("Enter a study term");
  }
}
```

Steps to Create and Run Java Code

- Right click on StudyPage.java in the package view.
- Choose Run As → Java Application.
- Save the class when prompted.
- Note results display in the Console View.



Implementing the Main Class Code

- Continue implementing the main class to accept a term, prompt for a definition, accept a definition, and finally, display the term and definition as shown below.

```
package studyTool;
import java.util.Scanner;

public class StudyPage {
   public static void main(String[] args) {

          Scanner scanterm = new Scanner(System.in);
          String termvar;
          System.out.println("Enter a study term");
          termvar = scanterm.nextLine();
          Scanner scandef = new Scanner(System.in);
          String termdef;
          System.out.println("Enter a definition");
          termdef = scandef.nextLine();

          System.out.println(termvar + ":   " + termdef);
     }
}
```

Implementing the Main Class Code

- You may have to correct some syntax errors that are a result of typing errors.
- See if you can correct them without asking for assistance.
- Pay particular attention to the ";" at the end of each line, and that your "{" (left curly brace) has a matching "}" (right curly brace.

Describe the StudyPage Class Code

- See if you can describe how the code in the StudyPage class is interpreted by Java.
- Add comments to your code to describe what you think the lines of code do.
  - Comments are ignored by the Java compiler.
  - To add a comment, type // in front of the comment.

Program to Convert Gallons to Liters

- What is the formula to convert gallons to liters?
  - 1 US gallon is equivalent to 3.78541178 Liters.

Pseudocode for Gallons to Liters Conversion

- What is the math calculation to complete the conversion?
- Pseudocode would look like:
  - Use Scanner to read in the number of gallons.
  - Store the number of gallons in a variable.
  - Multiply that variable by 3.785 (variable*3.785) and store this new value into a second variable (Hint: use double for the variables, not int).
  - Display the output.
  - Use the StudyPage program as a guide to create the program on your own.

Hints for Gallons to Liters Conversion

- Hints:
  - an int variable type is used to store whole numbers
  - a double variable type is used to store decimal numbers
  - int numgallons = 0; //declare a variable for number of gallons
  - double converttoliters = numgallons * 3.785

Terminology

Key terms used in this lesson included:

- Camel case
- Eclipse: edit and view areas, perspective, workspace
- Java JRE vs. Java JDK
- Java classes
- Java packages
- Java main methods

Summary

In this lesson, you should have learned how to:

- Identify components of Eclipse
- Identify components of a Java application
- Compile an application
- Test to ensure application is complete
- Write the code for GalToLit.java

Summary

In this lesson, you should have learned how to:

- Modify a program to execute error free
- Modify a program to use a formula to convert units of measure


ORACLE
ACADEMY