ORACLE

Academy

Java Fundamentals

4-1

Getting Started with Java

ORACLE
Academy

Objectives

- This lesson covers the following objectives:
  - Identify components of a Java IDE
  - Identify components of a Java application
  - Compile an application
  - Test to ensure application is complete
  - Write the code for GalToLit.java
  - Modify a program to execute error free
  - Modify a program to use a formula to convert units of measure

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 execute Java applications

The JRE is sometimes referred to as the VM (Virtual Machine). For this course, the JDK will be required.

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 installed on your computer

With appropriate arguments, java is the command that runs Java programs. javac is the command that compiles .java source code files into .class files that can then be executed with the java command. The IDE is the front end interface that uses these commands.

Steps to Launch Eclipse

- The steps shown in the following slides are for using the Eclipse Java IDE
- If you are using a different Java IDE, the steps and screens will be slightly different

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

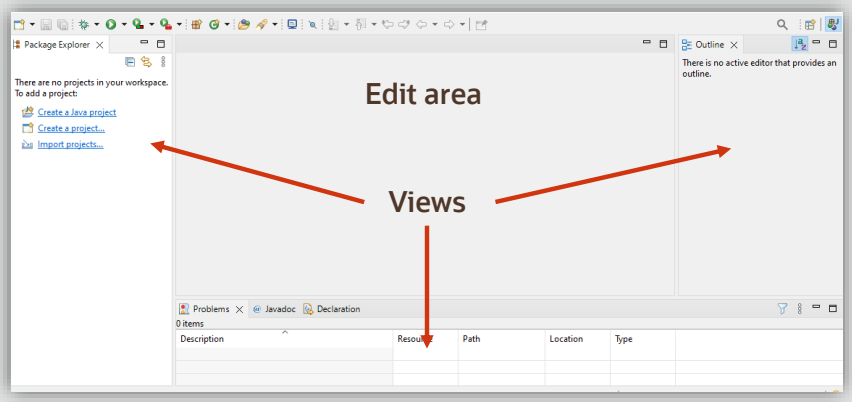
The screen captures are from Eclipse version 2021-12. The most recent version of Eclipse should be used with this course.

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



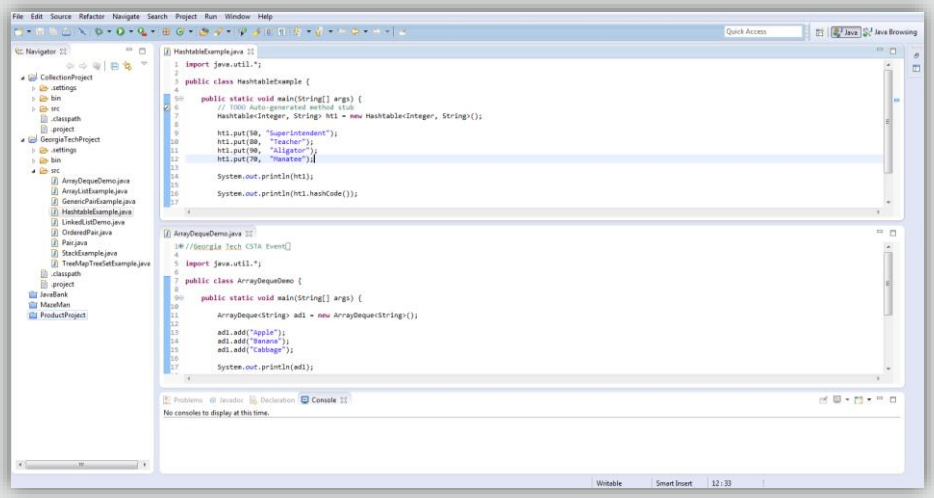
ORACLE Academy

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

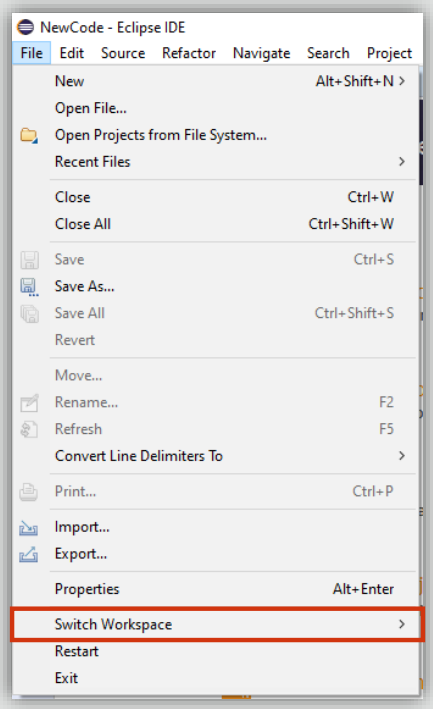
The Workspace

- 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

Do not keep the workspace in the same folder as Eclipse itself. That way if Eclipse is corrupted or upgraded the workspace will not be lost. Keeping both on a portable device such as a flash drive is an option. BACK UP THE WORKSPACE!

Switching Workspaces

- You can Switch Workspaces
  - From the File menu 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

Packages are covered in detail in Java Programming and are used to organize Java classes. A project may be created without using packages. Eclipse refers to this as the default package and warns that the practice is discouraged.

Projects in Eclipse

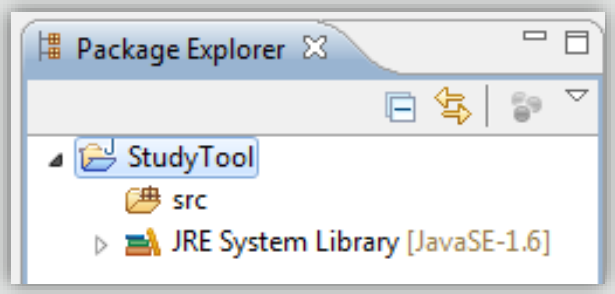
- In Eclipse:
  - All programs must reside inside a project for proper compilation
  - You may have one or multiple class files in one project
  - One of the classes must contain a main method

Create a Project in a Java IDE

- Create your first project in your Java IDE
  - The reference materials for this lesson show the steps to do this for both Eclipse and NetBeans
  - If you are using a different IDE, your instructor will demonstrate the required steps

Project Display

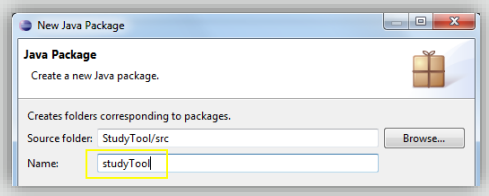
- The project is created and displayed as a folder
- It displays in the Package view to the left of the edit area

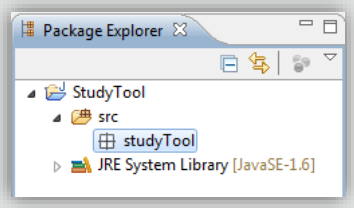


Naming Packages

- For this course, package names should be the same as the project name using lower camel case

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





Conventional naming practices for packages will be covered in Java Programming. For now, we use the same name for the package and the project for convenience.

A Java Class

- A class in Java is a construct that is used as a blueprint to create objects
- A class is also a construct in which objects are created

The Main Method

- A main method in Java is the method inside a class that runs when the class is compiled and executed
- The class containing the main method is also referred to as the Driver Class

Syntax Errors

- You may have to correct some syntax errors that are a result of typing errors
- Your IDE will highlight syntax errors in your code
- 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)

Java class comments

- Add comments to your code to describe what the lines of code do
  - Comments are ignored by the Java compiler
  - To add a comment, type **//** in front of the comment for a single line comment
  - To add a block comment (for multiple lines) use **/*** to begin the comment, and ***/** to end

Terminology

- Key terms used in this lesson included:
  - Camel case
  - Java IDE:
    - 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 a Java IDE
  - Identify components of a Java application
  - Compile an application
  - Test to ensure application is complete
  - Write the code for GalToLit.java
  - Modify a program to execute error free
  - Modify a program to use a formula to convert units of measure


ORACLE
Academy