

The background is a light blue to white gradient. It features several realistic water droplets of various sizes, some with highlights and shadows, scattered across the surface. A faint, large, light blue Java logo is centered in the background, behind the title text.

JAVA INTRODUCTION

大綱

◆基本介紹

- 起源
- 特性
- 應用

◆開發環境介紹

- 基本環境
- 開發工具IDE

◆ECLIPSE安裝與設定

JAVA是什麼?

- JAVA是美國昇陽公司(SUN MICROSYSTEMS)所開發的程式語言。
- JAVA是一種電腦程式設計語言，擁有跨平台、物件導向、自動垃圾回收等特性，廣泛應用於各式應用程式、**WEB**應用開發與行動應用開發。

JAVA與C++

- JAVA程式語言的風格十分接近C++語言。
 - 繼承了C++語言物件導向技術。
 - JAVA捨棄了C++語言中容易引起錯誤的指標，改以參照取代。
 - 移除C++的運算子多載。
 - 移除多重繼承特性，改用介面取代。
 - 增加垃圾回收器功能。
 - 在**JAVA SE 1.5**版本中引入了泛型程式設計、型別安全的列舉、不定長參數和自動裝/拆箱特性。
 - **JAVA 6, 7, 8, 9, 10**

JAVA 程式語言 (1/2)

- 昇陽電腦對JAVA語言的解釋是：「JAVA程式語言是個簡單、物件導向、分布式、解釋性、健壯、安全與系統無關、可移植、高效能、多執行緒和動態的語言」
- SUN公司在推出JAVA時就將其作為開放的技術。JAVA開發公司被要求所設計的JAVA軟體必須相互相容。
 - 這與微軟公司所倡導的注重精英和封閉式的模式完全不同。

JAVA 程式語言 (2/2)

- 後來微軟公司後來推出模仿**JAVA**的**C#**語言與**.NET**平台。後來**SUN**公司被甲骨文公司(**ORACLE**)併購，**JAVA**也隨之成為甲骨文公司的產品。
- 本課程用 **JAVA 8**

起源 (1/2)

- 1991年時，昇陽公司為了開發消費性電子產品的控制軟體而成立了一個成為(GREEN PROJECT)的研究計劃，而由於『當時』所使用的「C++」程式語言有以下缺點：
 - ✓ 太複雜以至很多開發者經常錯誤使用。
 - ✓ 缺少垃圾回收系統。
 - ✓ 移植的安全性低。
 - ✓ 分佈程式設計差。
 - ✓ 無多執行緒功能。

起源 (2/2)

- 當時的計畫主持人 **JAMES GOSLING** 便以「**C++**」為基礎，重新開發了一套新的程式語言，名稱為「**OAK**」，它便是 **JAVA** 的前身。



- **WEB**於1995年開始盛行，**JAVA**也隨之一炮而紅
- 2009年4月，**SUN**被**ORACLE**（甲骨文）公司併購



JAVA 特性

- 物件導向
- 跨平台性
- 自動垃圾回收

應用

- 目前根據**JAVA**使用的領域不同，又可分為以下版本：
 - JAVA PLATFORM, ENTERPRISE EDITION (JAVA EE)
 - JAVA PLATFORM, STANDARD EDITION (JAVA SE)
 - JAVA PLATFORM, MICRO EDITION (JAVA ME)
 - APPLET (WEB APPLICATION : CLIENT)
 - SERVLET (WEB APPLICATION : SERVER)
 - JAVASCRIPT (WEB SERVICE : CLIENT)
 - JSP (WEB SERVICE : SERVER)

大綱

◆ 基本介紹

- 起源
- 特性
- 應用

◆ 開發環境介紹

- 基本環境
- 開發工具IDE

◆ ECLIPSE安裝與設定

基本環境-API

- JAVA RUNTIME ENVIRONMENT(JRE)
- JAVA DEVELOPMENT KIT(JDK)

- 下載位置：

[HTTP://WWW.ORACLE.COM/TECHNETWORK/JAVA/JAVASE/DOWNLOADS/INDEX.HTML?SSSOURCEID=OTNJP](http://www.oracle.com/technetwork/java/javase/downloads/index.html?sssourceid=otnjp)

- JAVA SE 8U181 (AS OF 2018.08.02)

基本環境-下載JDK過程

Java SE Development Kit 7u51

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	67.7 MB	jdk-7u51-linux-arm-vfp-hflt.tar.gz
Linux ARM v6/v7 Soft Float ABI	67.68 MB	jdk-7u51-linux-arm-vfp-sflt.tar.gz
Linux x86	115.65 MB	jdk-7u51-linux-i586.rpm
Linux x86	132.98 MB	jdk-7u51-linux-i586.tar.gz
Linux x64	116.96 MB	jdk-7u51-linux-x64.rpm
Linux x64	131.8 MB	jdk-7u51-linux-x64.tar.gz
Mac OS X x64	179.49 MB	jdk-7u51-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.02 MB	jdk-7u51-solaris-i586.tar.Z
Solaris x86	95.13 MB	jdk-7u51-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.53 MB	jdk-7u51-solaris-x64.tar.Z
Solaris x64	16.28 MB	jdk-7u51-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	139.39 MB	jdk-7u51-solaris-sparc.tar.Z
Solaris SPARC	98.19 MB	jdk-7u51-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u51-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.33 MB	jdk-7u51-solaris-sparcv9.tar.gz
Windows x86	123.64 MB	jdk-7u51-windows-i586.exe
Windows x64	125.46 MB	jdk-7u51-windows-x64.exe

基本環境-下載JDK過程

Java Platform, Standard Edition

Java SE 7u51

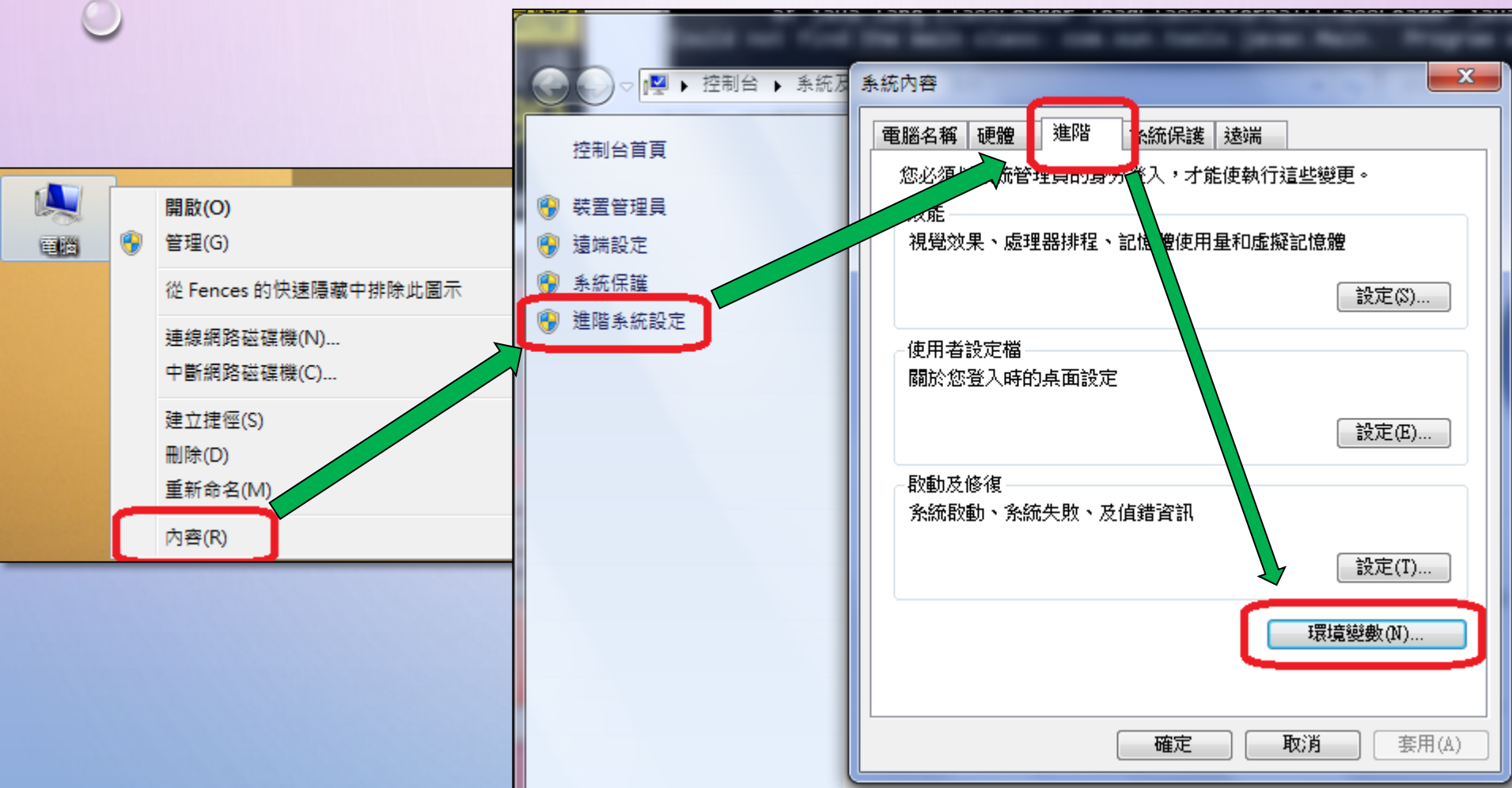
This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release.
[Learn more](#)

Which Java package do I need?

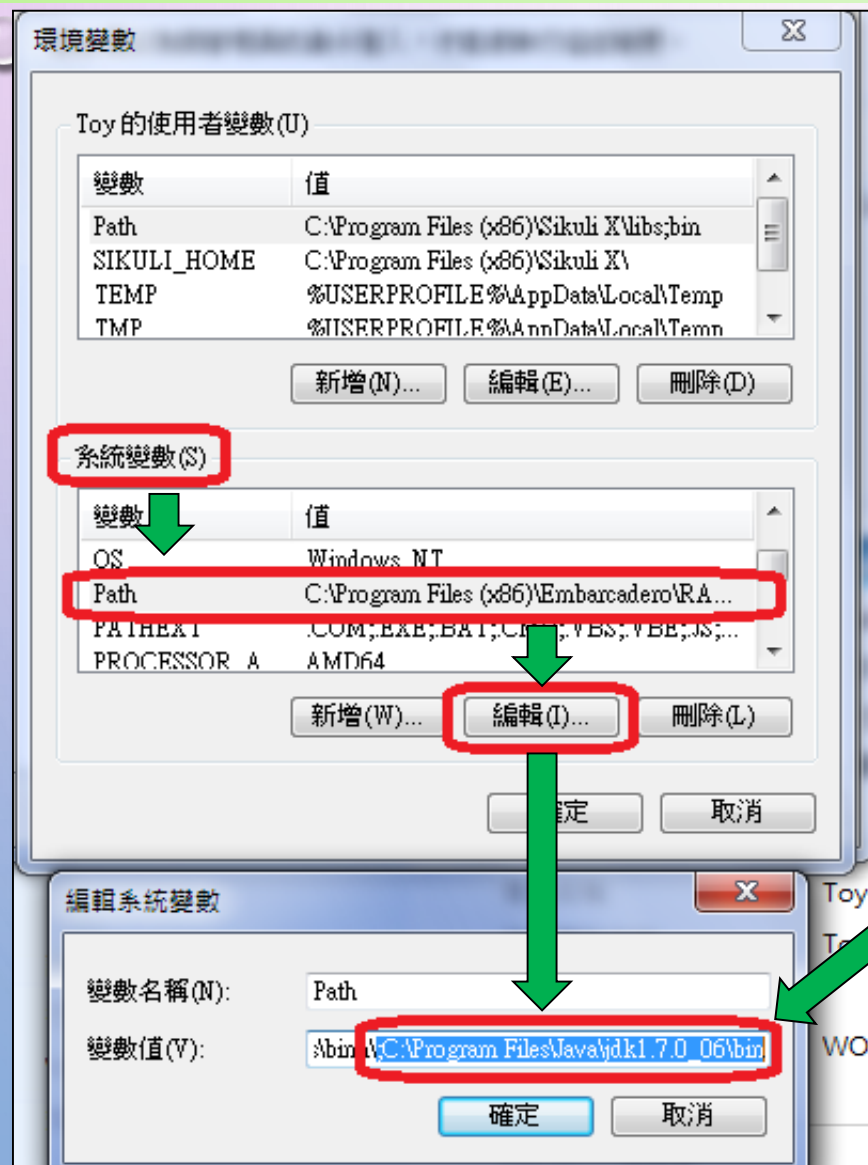
- **JDK:** (Java Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- **Server JRE:** (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. [Learn more](#)
- **JRE:** (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

<div>JDK</div> <div>DOWNLOAD</div> <div>JDK 7 Docs<ul style="list-style-type: none">▪ Installation Instructions▪ ReadMe</div>	<div>Server JRE</div> <div>DOWNLOAD</div> <div>Server JRE 7 Docs<ul style="list-style-type: none">▪ Installation Instructions▪ ReadMe</div>	<div>JRE</div> <div>DOWNLOAD</div> <div>JRE 7 Docs<ul style="list-style-type: none">▪ Installation Instructions▪ ReadMe</div>
---	---	---

基本環境-編譯前設定(1/2)



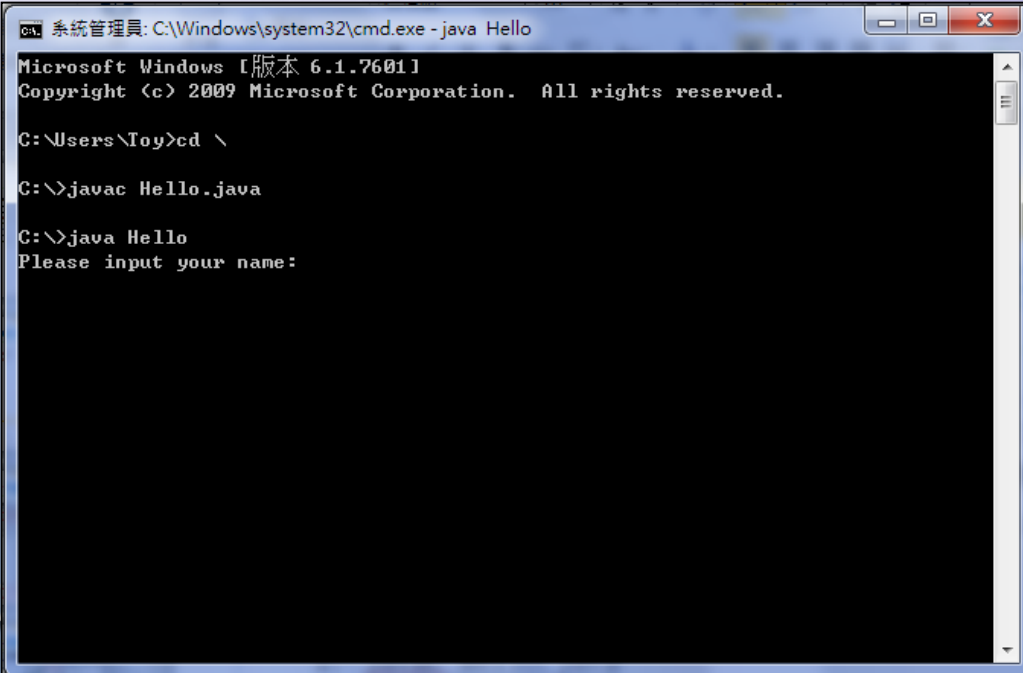
基本環境-編譯前設定(2/2)



在這加入 “;C:\JDK安裝位置\bin”
通常完整路徑為
C:\Program Files\Java\jdk1.x.x\bin
PS.整個路徑前一定要加個“;”

基本環境-第一次編譯執行

- STEP 1: 新增一個”純文字筆記本”於C槽根目錄(C:\)內。
- STEP 2: 複製簡報附的簡易程式碼，貼入，並存檔為**HELLO.JAVA**
- STEP 3: 打開”命令提示字元(開啟列->附屬應用程式->命令提示字元”
- STEP 4: 依序打入以下指令，打完記得按下“ENTER鍵”(如下圖)
 1. CD \
 2. JAVAC HELLO.JAVA
 3. JAVA HELLO



```
系統管理員: C:\Windows\system32\cmd.exe - java Hello
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Ioy>cd \

C:\>javac Hello.java

C:\>java Hello
Please input your name:
```

開發工具IDE

- 整合開發環境（INTEGRATED DEVELOPMENT ENVIRONMENT, IDE)是一種輔助程式開發人員開發軟體的應用軟體。
- 目前JAVA IDE主流為兩種工具
 - ECLIPSE (IBM)
 - INTELLIJ
 - NETBEAN (SUN)

	優點	缺點
Eclipse	輕巧、執行較不耗資源、google主要使用	外掛套件太多
NetBean	操作直覺	執行耗資源

大綱

◆ 基本介紹

- 起源
- 特性
- 應用

◆ 開發環境介紹

- 基本環境
- 開發工具IDE

◆ ECLIPSE安裝與設定

ECLIPSE下載

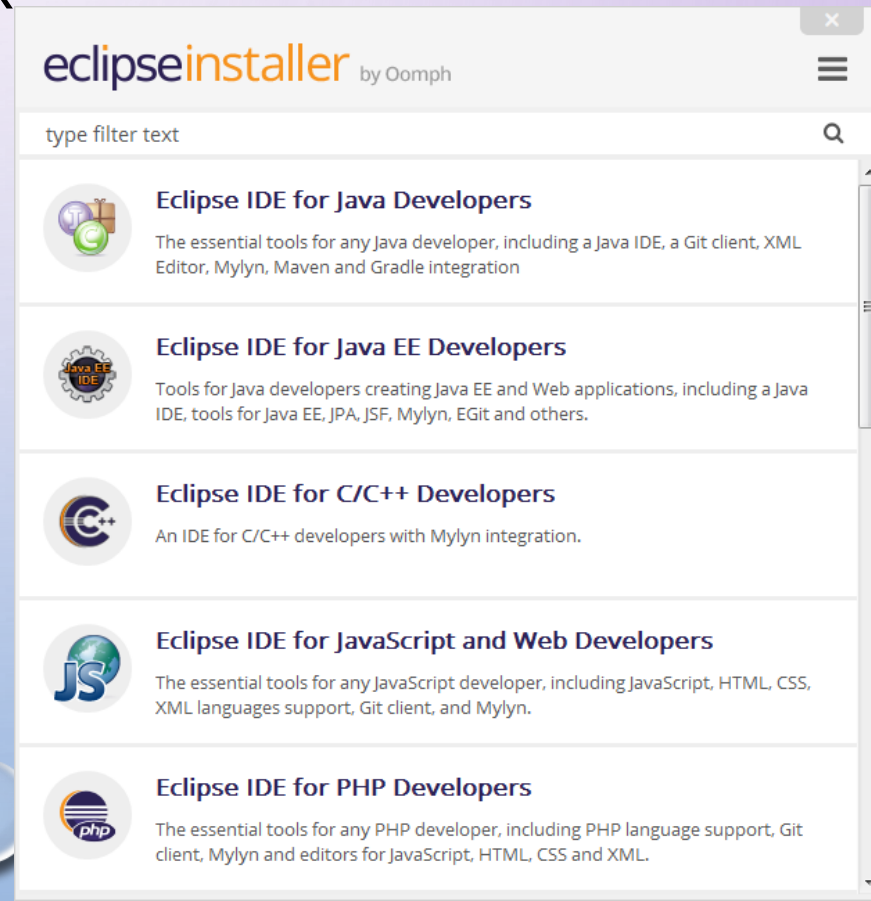
- ECLIPSE 下載位置:

[HTTP://WWW.ECLIPSE.ORG/DOWNLOADS](http://www.eclipse.org/downloads)

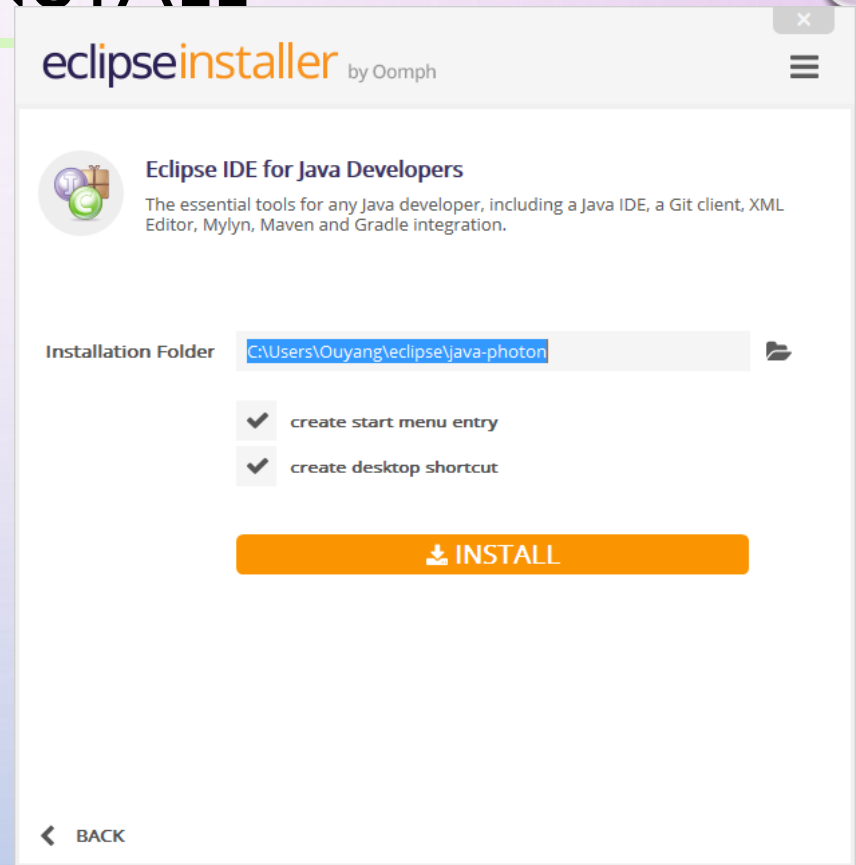
- DOWNLOAD ECLIPSE INSTALLER

- ECLIPSE-INST-WIN64.EXE

- EXECUTE ECLIPSE INSTALLER



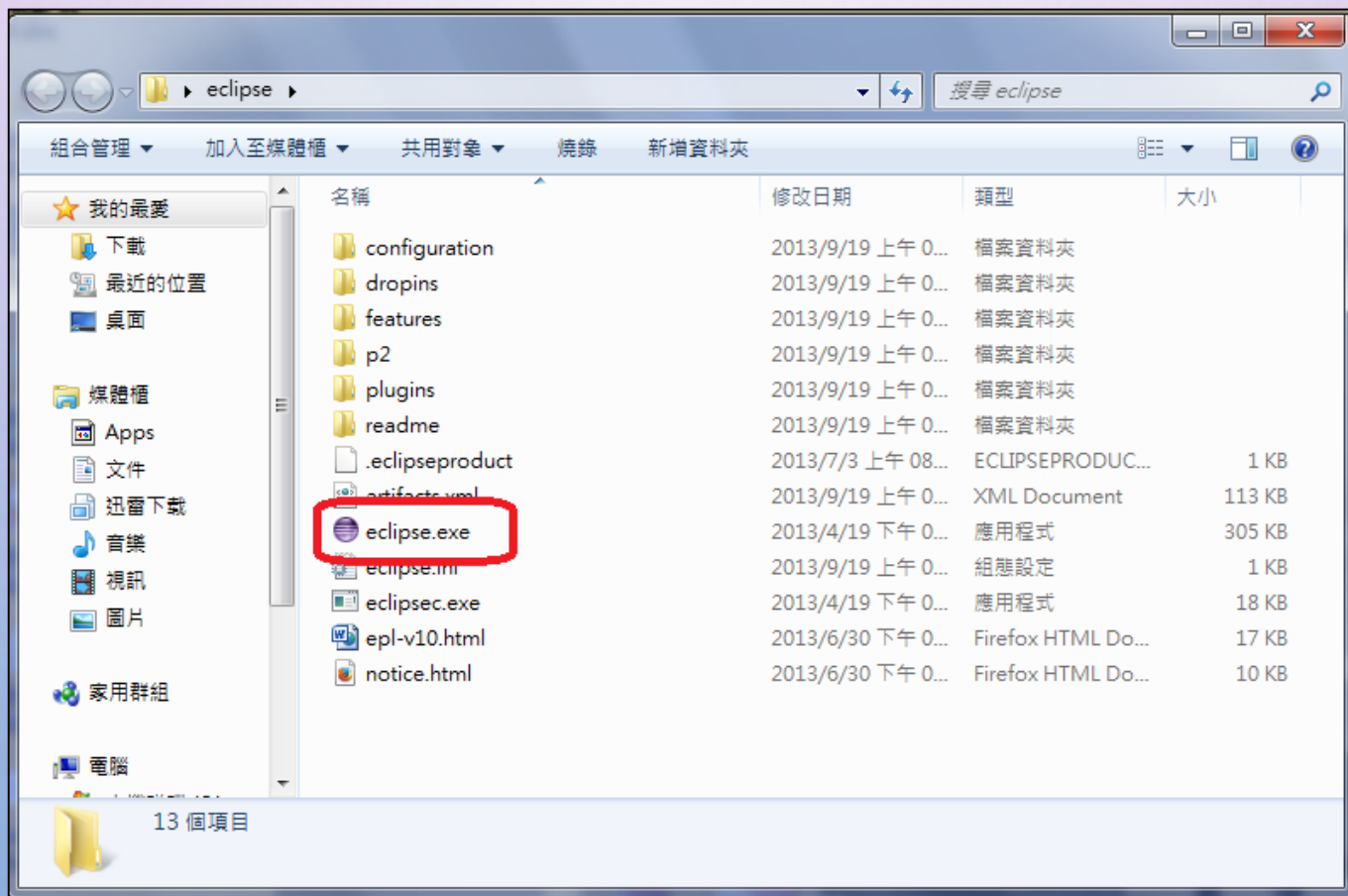
CLICK INSTALL



- ACCEPT LICENSE (WAIT FOR INSTALLATION TO BE COMPLETE)
- CLICK “OPEN IN SYSTEM EXPLORER”
- CLICK LAUNCH (OR DOUBLE CLICK ECLIPSE.EXE)

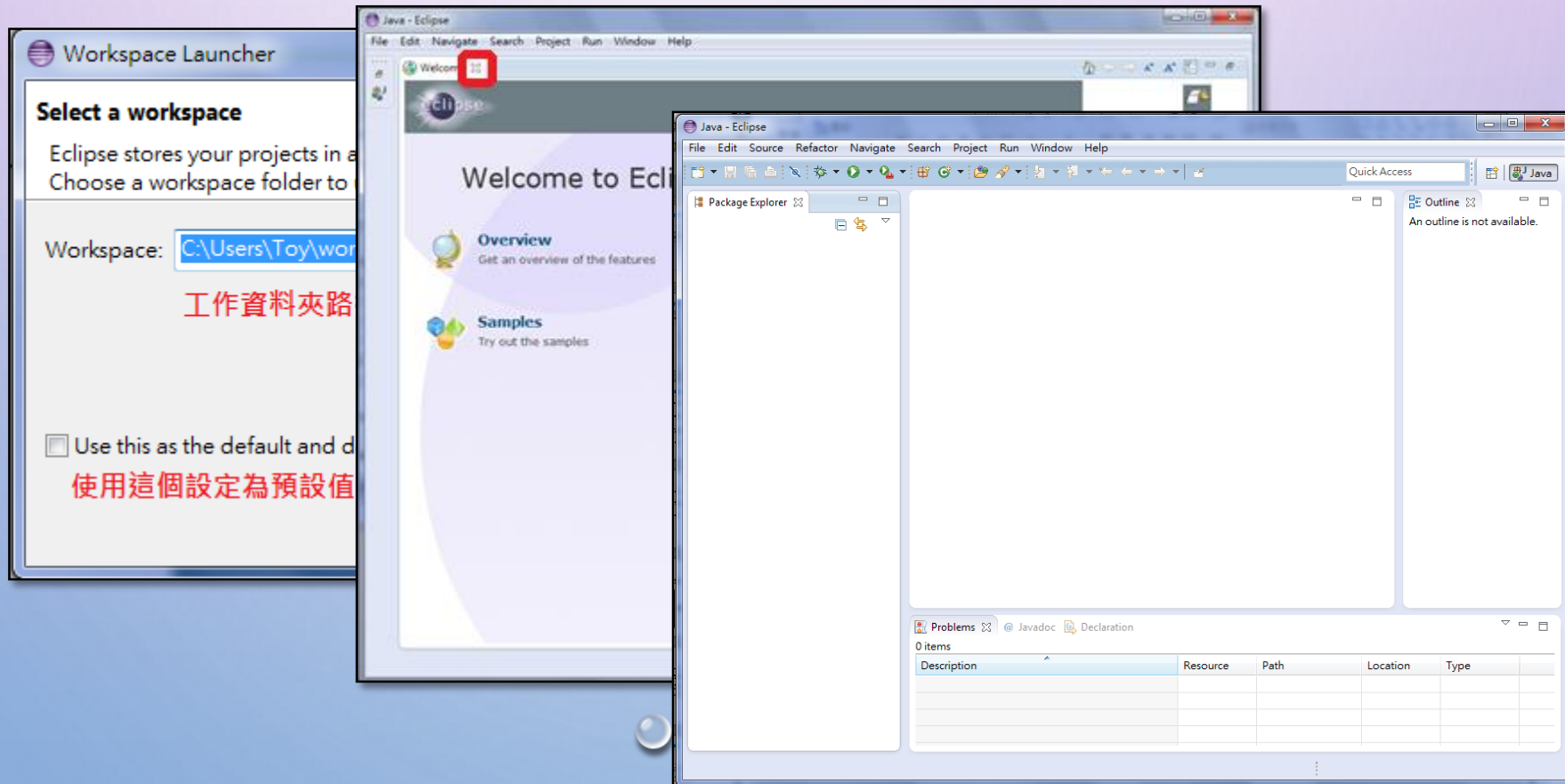
ECLIPSE安裝與使用

- 不用安裝，直接解壓縮即可使用。
- 執行檔為ECLIPSE.EXE



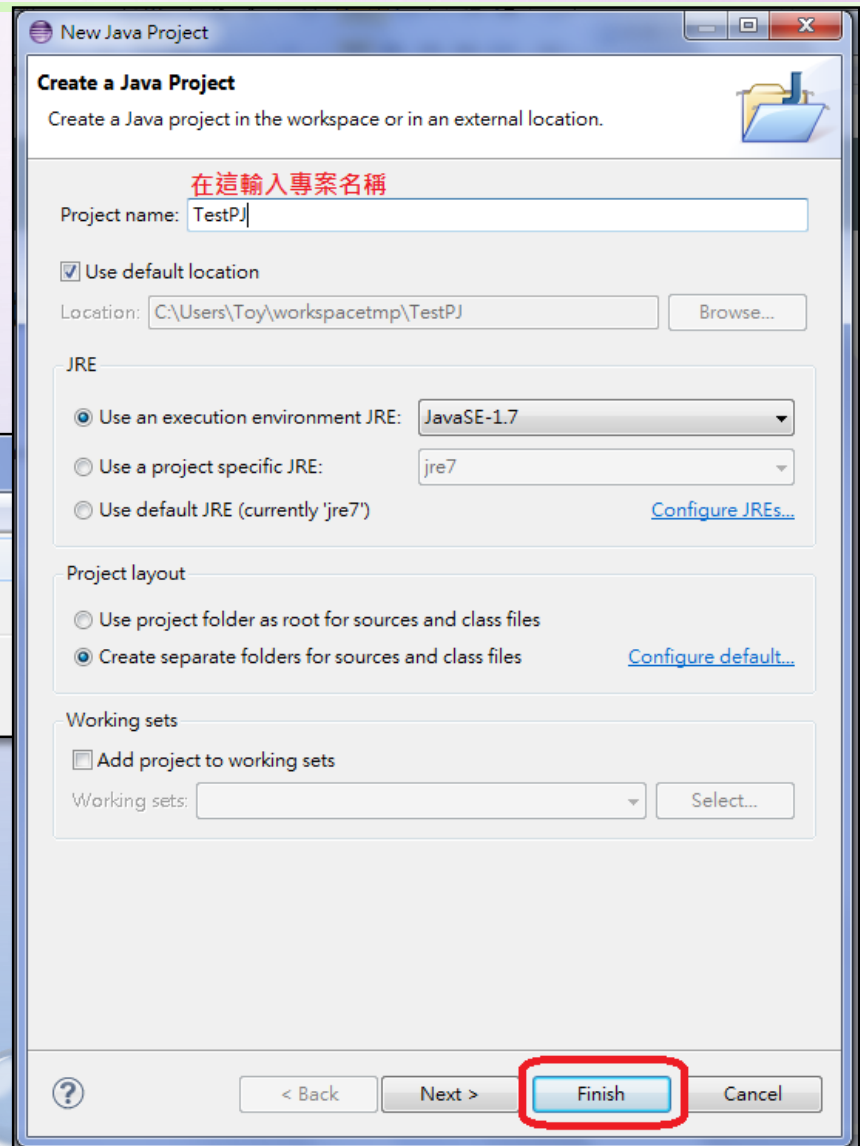
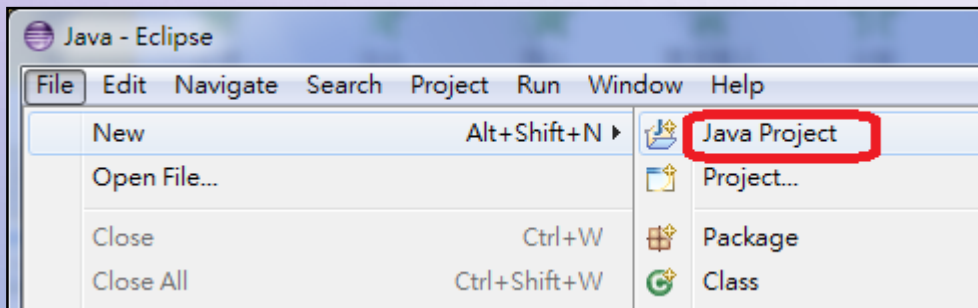
使用ECLIPSE編譯JAVA(1 / 3)

- 第一次開啟會顯示以下視窗，這是讓你選擇你的工作資料夾，以後建立的專案會放置於該資料夾內。
- 關閉開始頁(點擊圖中紅框的叉叉)，切換到工作環境。



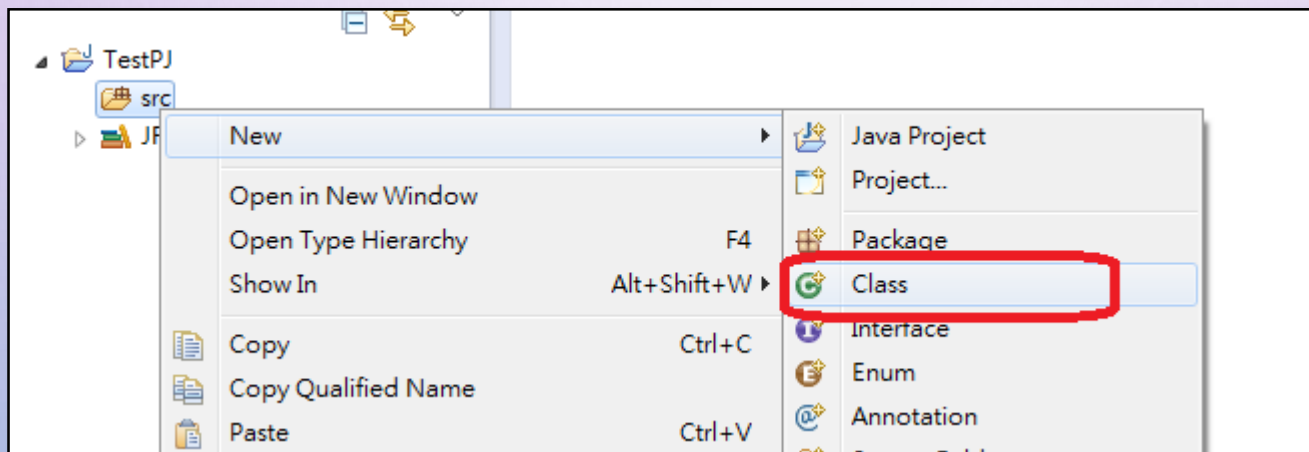
使用ECLIPSE編譯JAVA(2/3)

- 新建一個JAVA專案
- 打入你要建立的專案名，然後



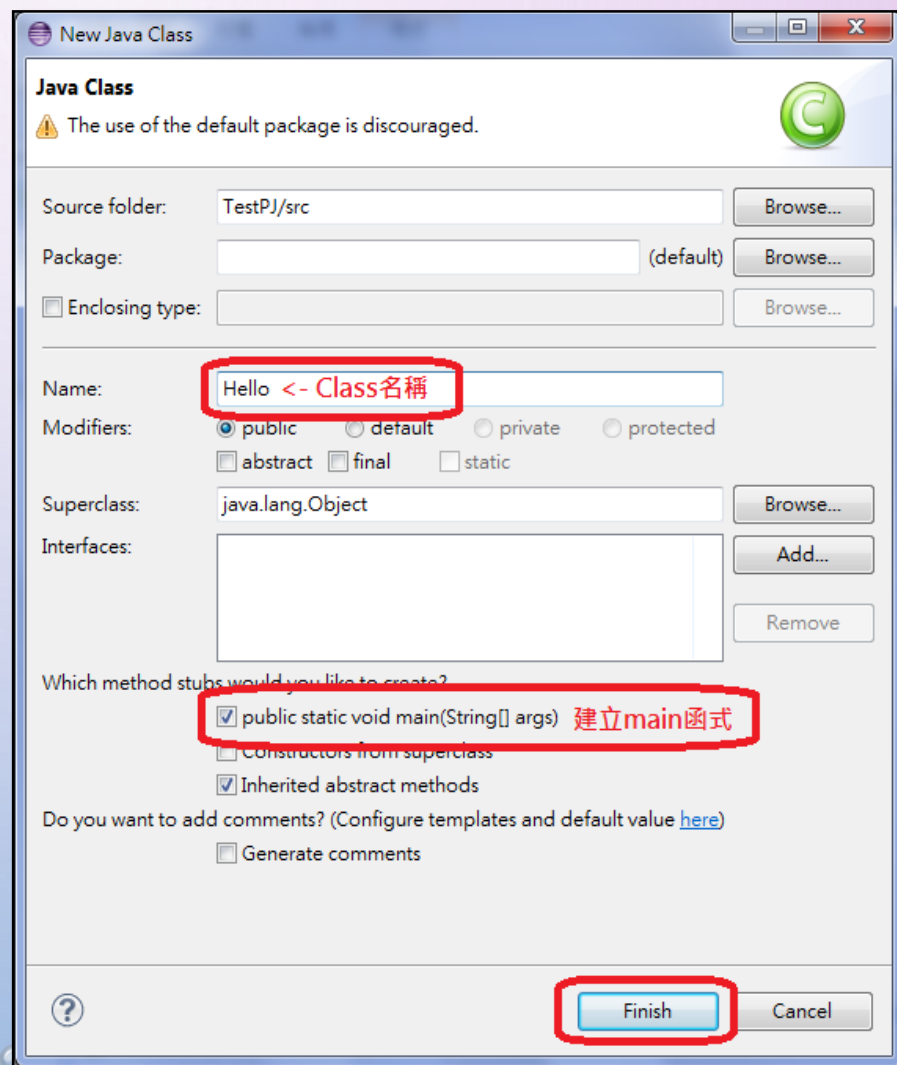
使用ECLIPSE編譯JAVA(2/3)

- 到工作區域左邊展開你新建的專案，到SRC資料夾上按下右鍵，點選 NEW->CLASS



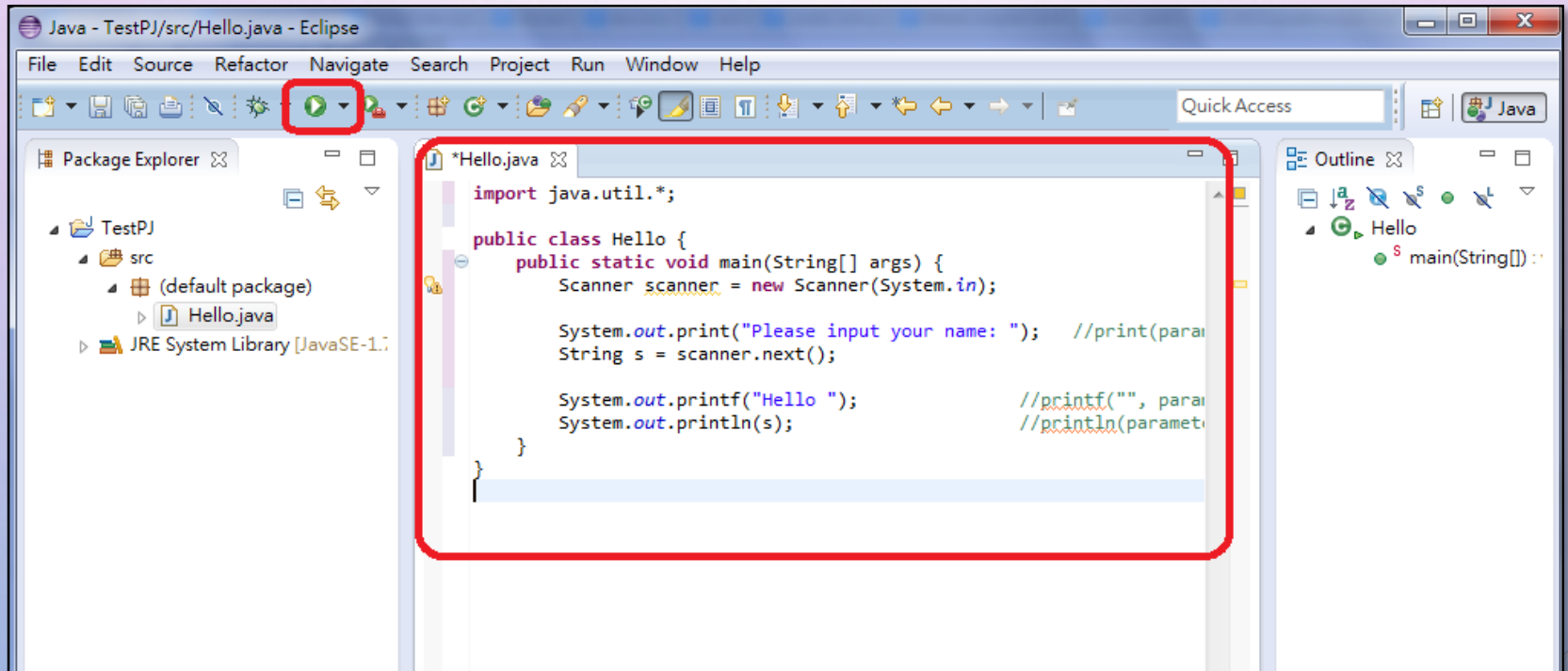
使用ECLIPSE編譯JAVA(2/3)

- 輸入CLASS名稱，勾選中間紅框的勾，點擊FINISH按鈕。
(PS.因為是整個專案的第一份程式碼，所以需要一個MAIN，才會需要勾選該選項)



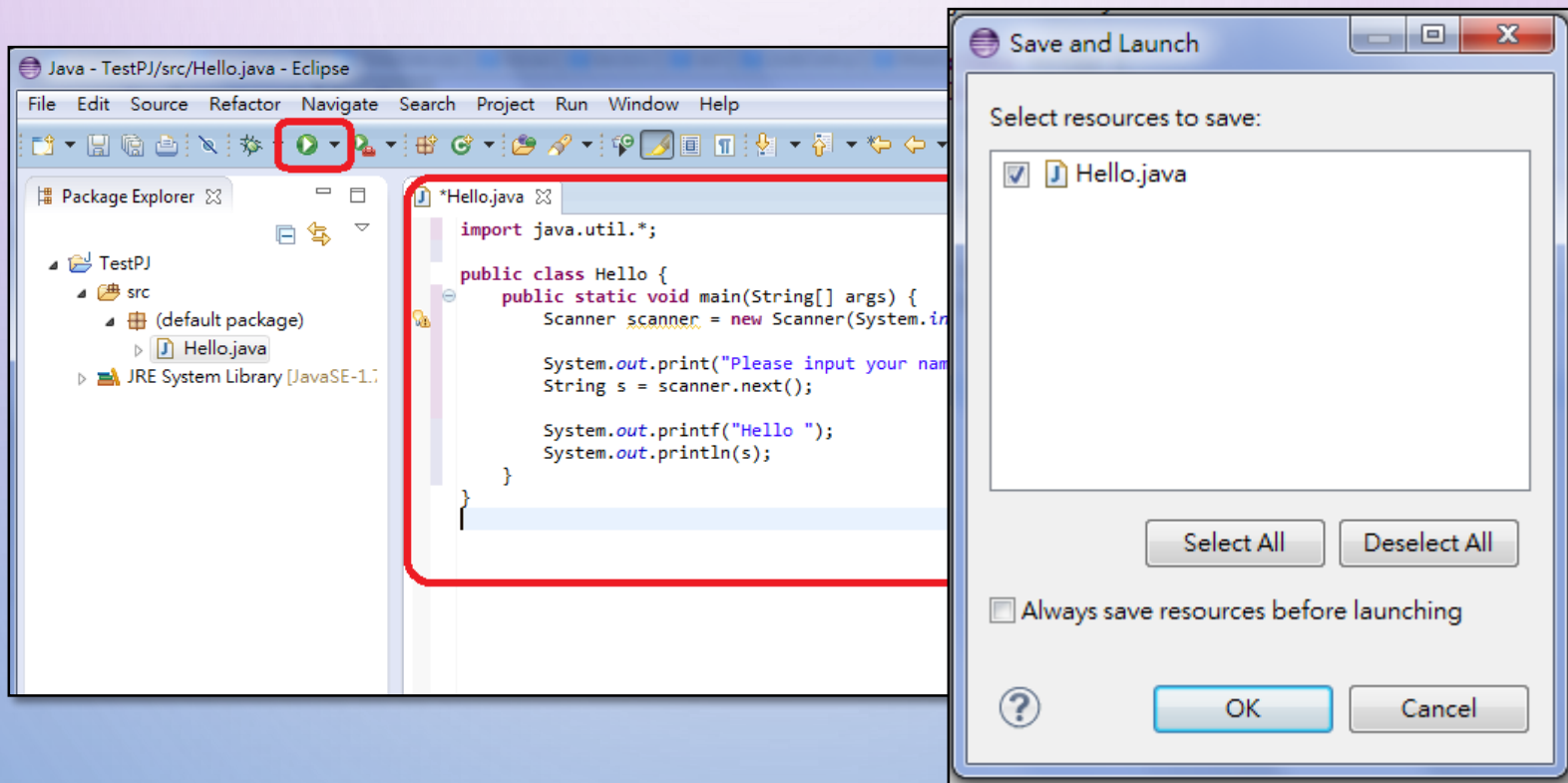
使用ECLIPSE編譯JAVA(3/3)

- STEP 1: 複製簡報附的簡易程式碼，貼入中間紅框處。



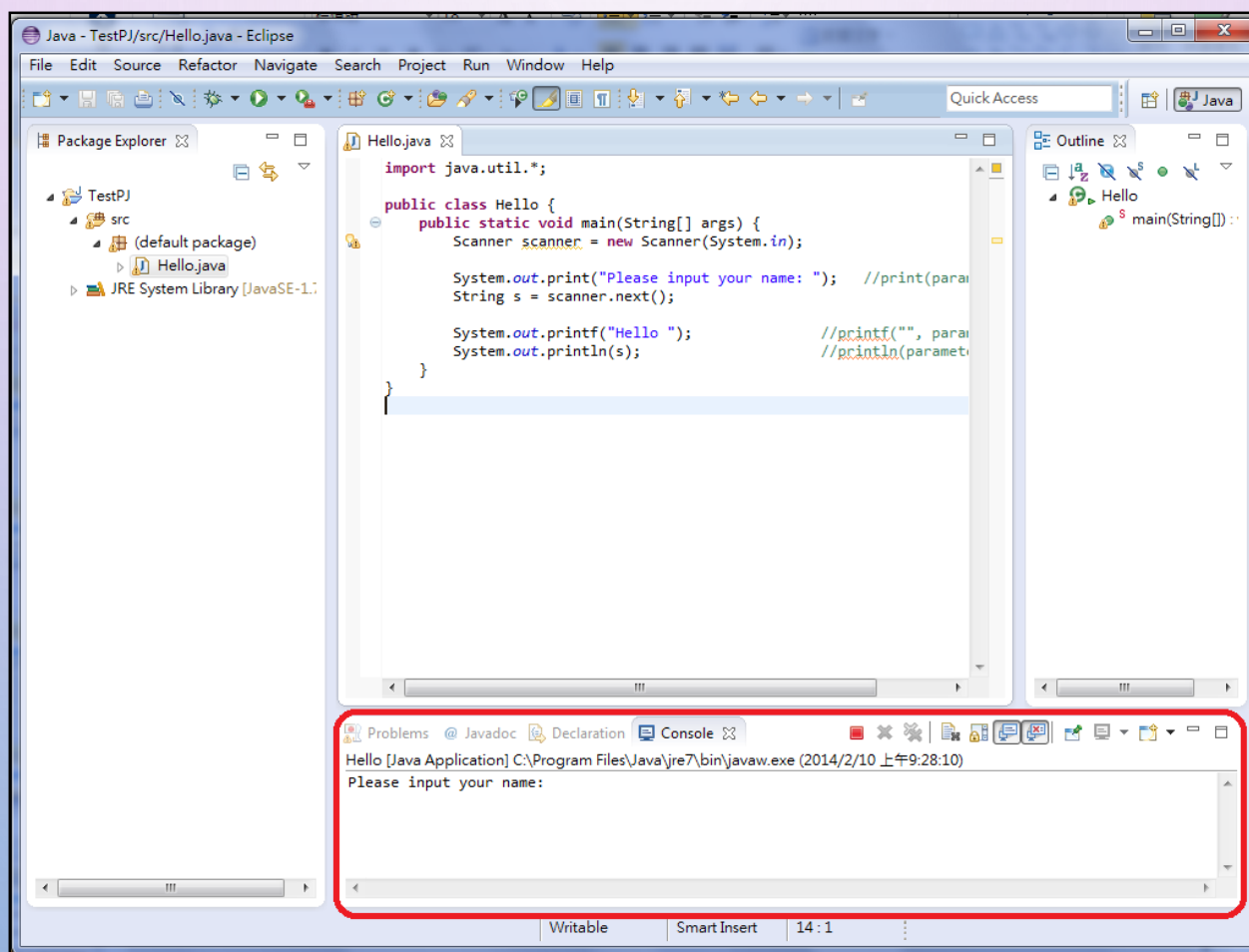
使用ECLIPSE編譯JAVA(3/3)

- STEP 2: 點極左上紅框，開始編譯程式碼並執行，點擊後可能會彈出另一個視窗，這是問你是否要儲存程式碼。直接點擊OK按鈕。



使用ECLIPSE編譯JAVA(3/3)

- STEP 3: 執行結果顯示於工作區下方的CONSOLE區域，如下圖紅框內。





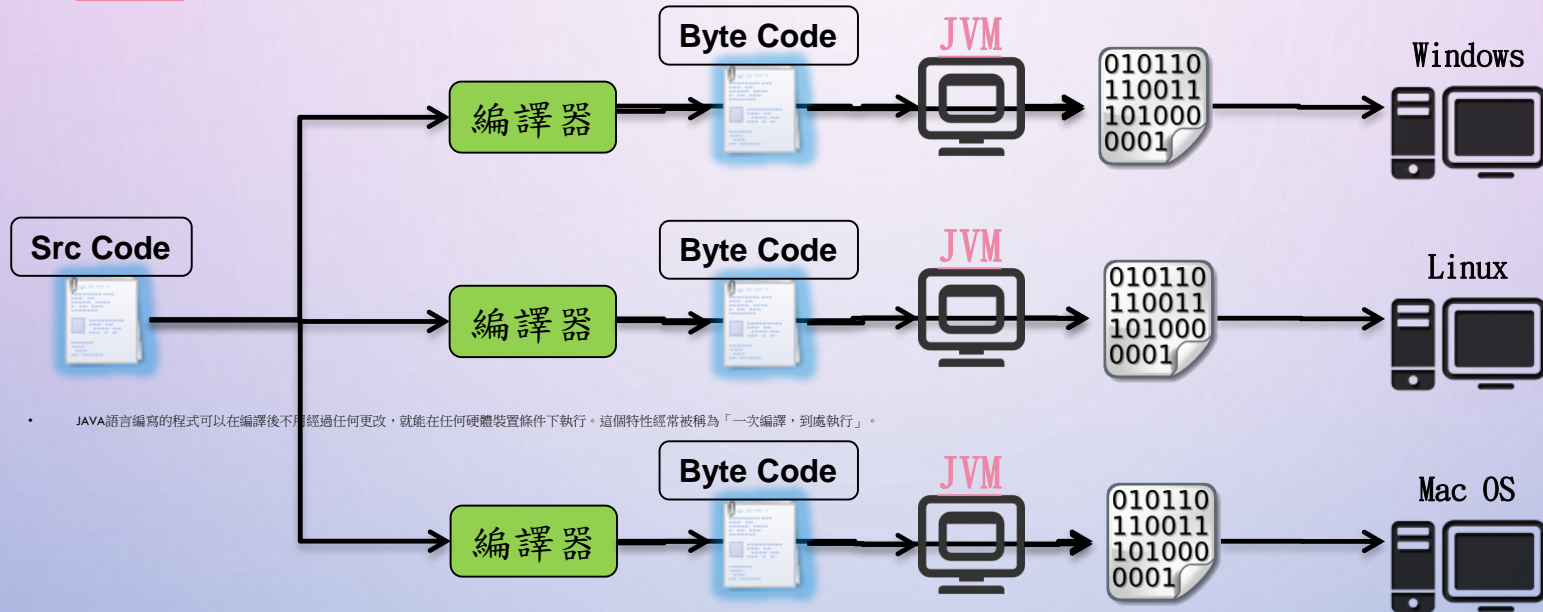
— END —

物件導向

- 一種程式設計典範，同時也是一種程式開發的方法。
- 物件(OBJECT)指的是類別(CLASS)的例項 (INSTANCE)。它將物件作為程式的基本單元，將程式和資料封裝其中，以提高軟體的重用性、靈活性和擴充功能性。
- 物件導向三大概念：封裝、繼承、多型

跨平台性

程式碼的運行方式



JAVA語言編寫的程式可以在編譯後不經過任何更改，就能在任何硬體裝置條件下執行。這個特性經常被稱為「一次編譯，到處執行」。

JAVA VIRTUAL MACHINE (JVM)

- JVM是一種抽象化的電腦，透過在實際的電腦上模擬模擬各種電腦功能來實作的。
- JAVA虛擬機器有自己完善的硬體架構，如處理器、堆疊、暫存器等，還具有相應的指令系統。
- JVM遮蔽了與具體作業系統平台相關的訊息，使得JAVA程式只需生成在JAVA虛擬機器上執行的目的碼（位元組碼），就可以在多種平台上不加修改地執行。

程式碼的運行

- 程式碼給電腦運行目前有三種方式：
 - 編譯式：原始的程式碼 (SOURCE CODE) 會經由編譯器 (COMPILER) 轉換成目的碼 (OBJECT CODE)，再由連結器 (LINKER) 轉換成可執行的二進位碼。
 - 組譯式：將組合語言的程式碼經過組譯器 (ASSEMBLER) 轉成目的碼 (OBJECT CODE)，再由連結器 (LINKER) 轉換成可執行的二進位碼。
 - 直譯式：原始的程式碼經過直譯器 (INTERPRETER) 轉換成可執行碼，在執行時才會將原始碼直譯成執行碼。

	語言層次	獨立運行	執行效能	編譯時間	使用語言
編譯	低階	可	高	有	C/C++、C#、JAVA……
組譯	高階	可	高	有	組合語言
直譯	高階	不可	低	無	BASIC、PHP、JAVA Script……

自動垃圾回收

- C++語言被用戶詬病的原因之一是大多數C++編譯器不支援垃圾收集機制。
- 通常使用C++編程的時候，程式設計師於程式中初始化對象時，會在主機記憶體分配一塊記憶體與位址，當不需要此對象時再釋放分配的記憶體位址。
- 而程式員忘記進行釋放，那麼就會造成記憶體洩漏（**MEMORY LEAK**）。
 - 長此以往，程式運行的時候可能會生成很多不清除的垃圾，浪費了不必要的記憶體空間。
 - 而且如果同一記憶體位址被釋放兩次的話，程式會變得不穩定，甚至程式或系統崩潰。

簡易程式碼範例

```
• import java.util.*;  
•  
• public class Hello {  
•     public static void main(String[] args) {  
•         Scanner scanner = new Scanner(System.in);  
•  
•         System.out.print("Please input your name: ");    //print(parameter)  
•         String s = scanner.next();  
•  
•         System.out.printf("Hello ");                    //printf("", parameter....)  
•         System.out.println(s);                          //println(parameter);  
•     }  
• }
```