

# Java程式設計第一類接觸

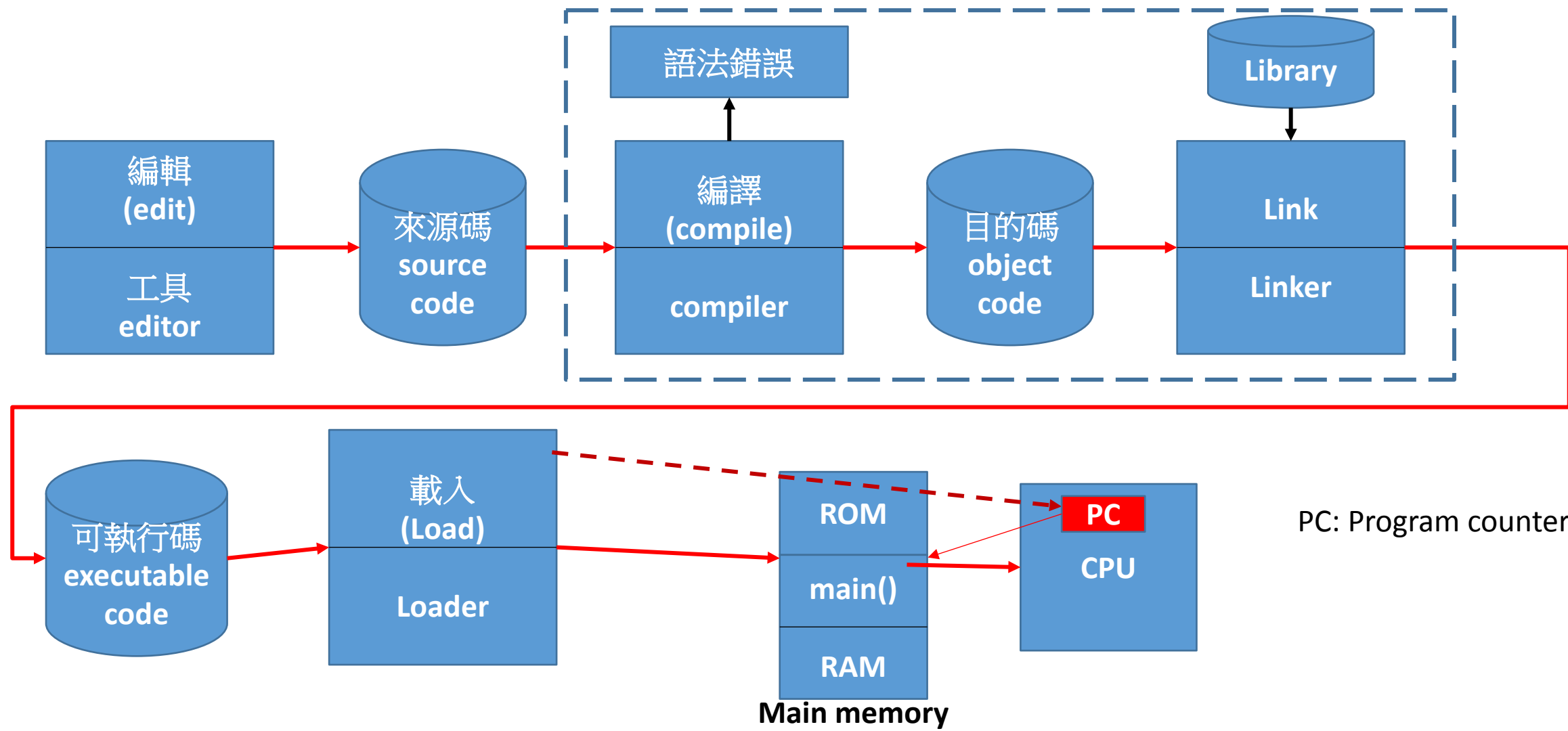
臺北市立大學 [資訊科學系\(含碩士班\)](#)

賴阿福

laiahfur@gmail.com

CS TEAM

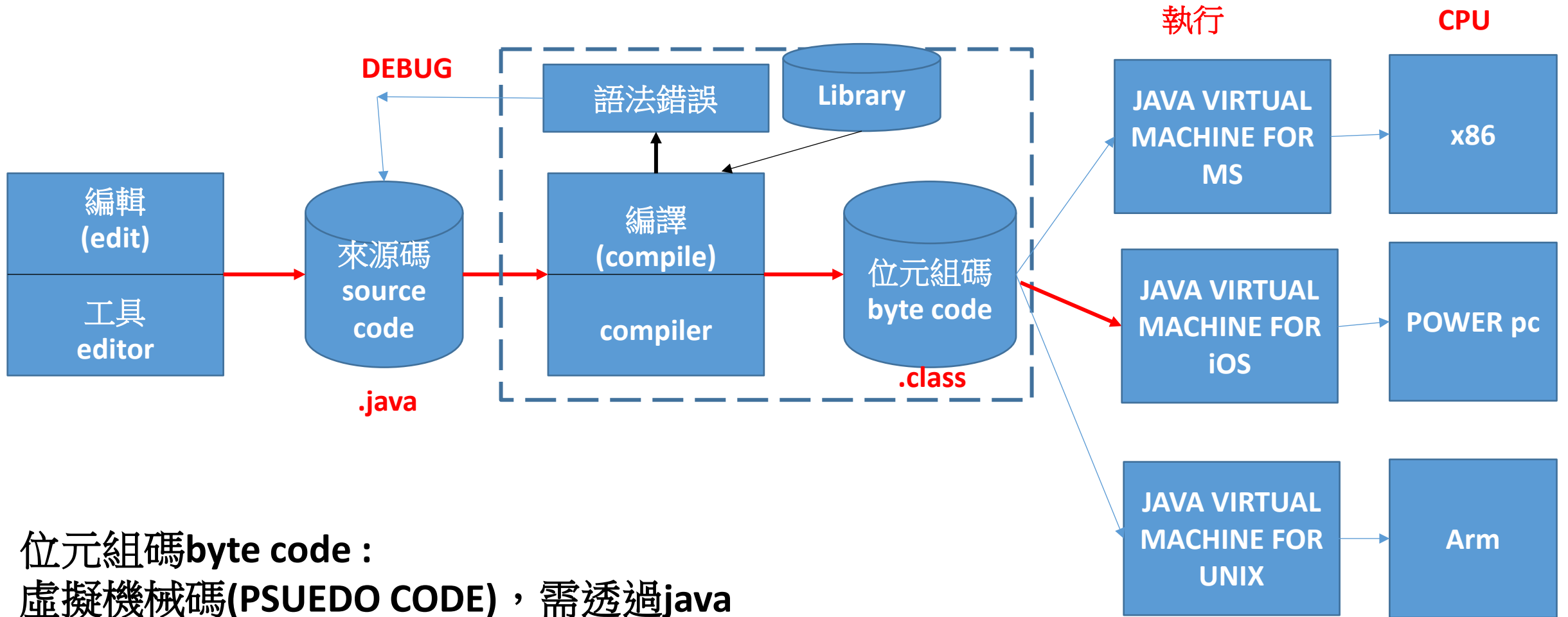
# 如何將高階語言轉換(翻譯;解譯;編譯;translate)為機械語言?



# Java電腦語言歷史

- **Java** (爪哇島) is an [island](#) of [Indonesia](#).
- **Java** is a general-purpose [computer programming language](#) that is [concurrent](#), [class-based](#), [object-oriented](#), and specifically designed to have as few implementation dependencies as possible.
- It is intended to let application developers “[write once, run anywhere](#)” (WORA), meaning that [compiled](#) Java code can run on all platforms that support Java without the need for recompilation.
- Java applications are typically compiled to **bytecode** that can run on any [Java virtual machine](#) (JVM) regardless of [computer architecture](#) (X86, MIPS, ARM....).
- Java was originally developed by [James Gosling](#) at [Sun Microsystems](#) (which has since been [acquired by Oracle Corporation](#)) and released in 1995 as a core component of Sun Microsystems' [Java platform](#).
- [James Gosling](#), Mike Sheridan, and [Patrick Naughton](#) initiated the Java language project in June 1991.
- The language derives much of its [syntax](#) from [C](#) and [C++](#), but it has fewer [low-level](#) facilities than either of them.
- The original and [reference implementation](#) Java [compilers](#), virtual machines, and [class libraries](#) were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the [Java Community Process](#), Sun [relicensed](#) most of its Java technologies under the [GNU General Public License](#). Others have also developed alternative implementations of these Sun technologies, such as the [GNU Compiler for Java](#) (bytecode compiler), [GNU Classpath](#) (standard libraries), and [IcedTea-Web](#) (browser plugin for applets).
- Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The language was initially called **Oak** after an [oak](#) tree that stood outside Gosling's office. Later the project went by the name *Green* and was finally renamed *Java*, from **Java coffee**. Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.
- [Sun Microsystems](#) (昇陽公司) released the first public implementation as Java 1.0 in 1995. The latest version is [Java 8](#) which is the [only version currently supported for free by Oracle](#) (甲骨文公司) .
- 資料來源 [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

# 如何將java語言轉換(翻譯;解譯;編譯;translate)為bytecode?

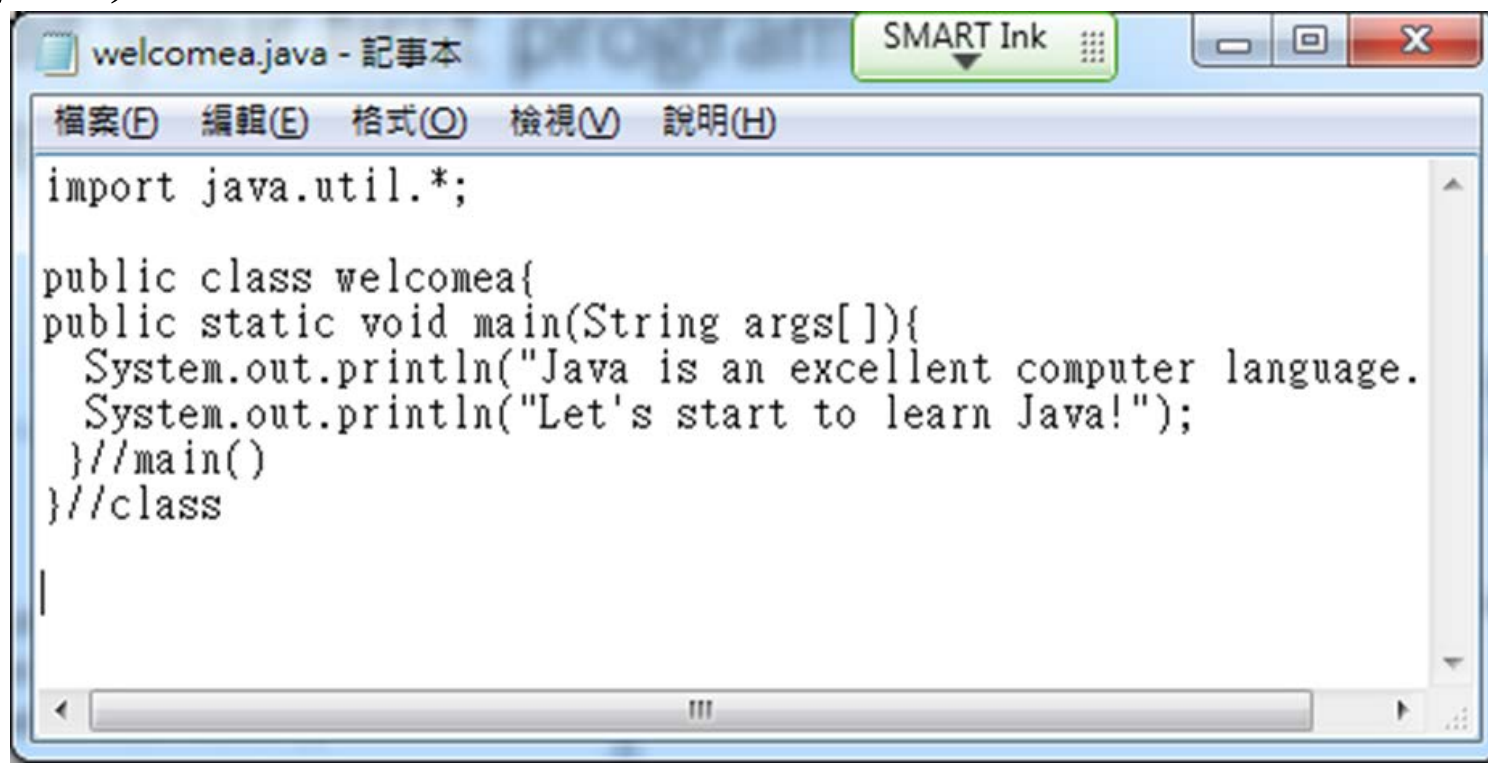


位元組碼byte code :  
虛擬機械碼(PSEUDO CODE) , 需透過java  
解譯程式才能執行

- write once, run anywhere
- 跨平台程式語言

# 語言？電腦語言？

- 語法：規則、結構（限制）
- 語句：statement
- 語意：解決方法（演算法）



The screenshot shows a window titled 'welcomea.java - 記事本' (welcomea.java - Notepad). The menu bar includes '檔案(F)', '編輯(E)', '格式(O)', '檢視(V)', and '說明(H)'. The code content is as follows:

```
import java.util.*;

public class welcomea{
    public static void main(String args[]){
        System.out.println("Java is an excellent computer language.
        System.out.println("Let's start to learn Java!");
    }//main()
}//class
```

# 學習程式之觀念

- 漸進式學習
- From unclear to clear
- 不用全部學完語法再開始程式設計
  - 不可能一步登天
- 從基本功能進化、優化
- 養成規劃、設計習慣
- 不要怕出現Bug
- 思考、動手寫程式

# 撰寫java程式三部曲

- 編輯(edit):如用記事本編寫
- 編譯(compile)
  - 如果編譯未完成(有錯誤:bug, error)
    - 除錯(debug): 找出錯誤，修正
    - 重新編譯
  - 如果編譯完成,產出 **.class**
- 執行 (run, execute) :如果編譯完成沒錯誤

你的第一支java程式

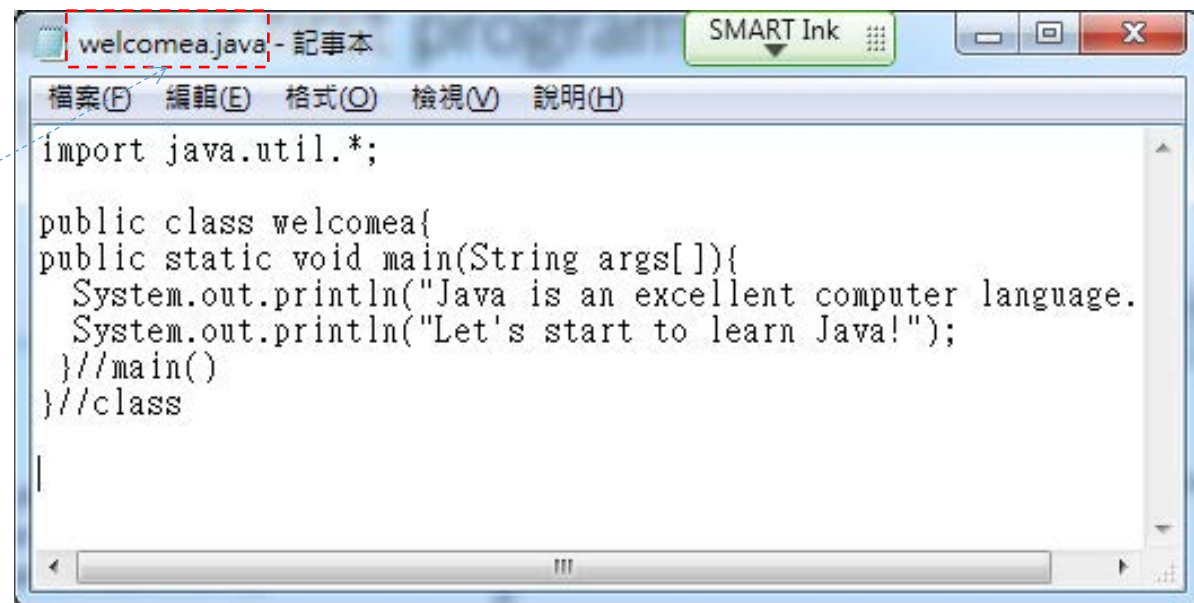


# How to edit your first java program?

- 新增“資料夾”存放java程式檔，自己為資料夾取名字(如afjava)
- 開啟記事本軟體
- 輸入第一個程式或由酷課雲下載welcomea.java至自己資料夾(如afjava)

```
import java.util.*;

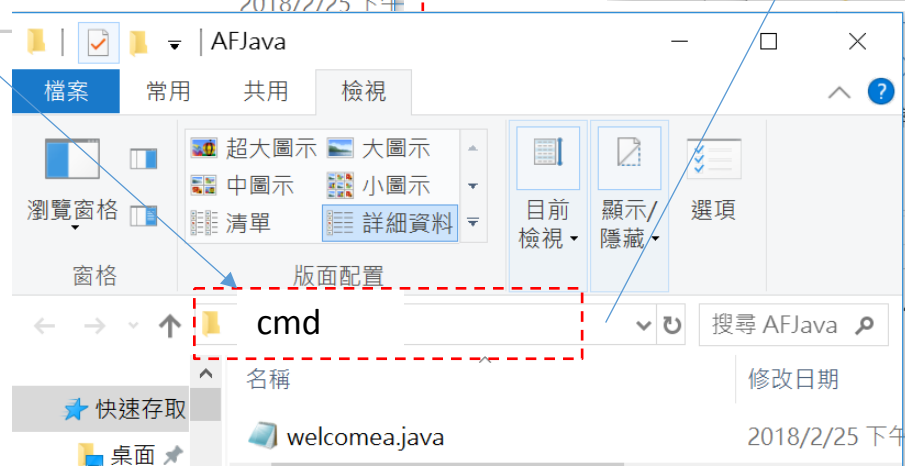
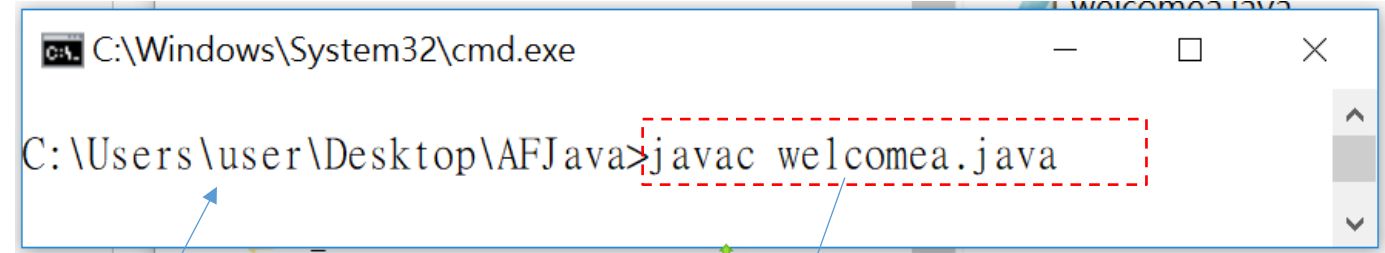
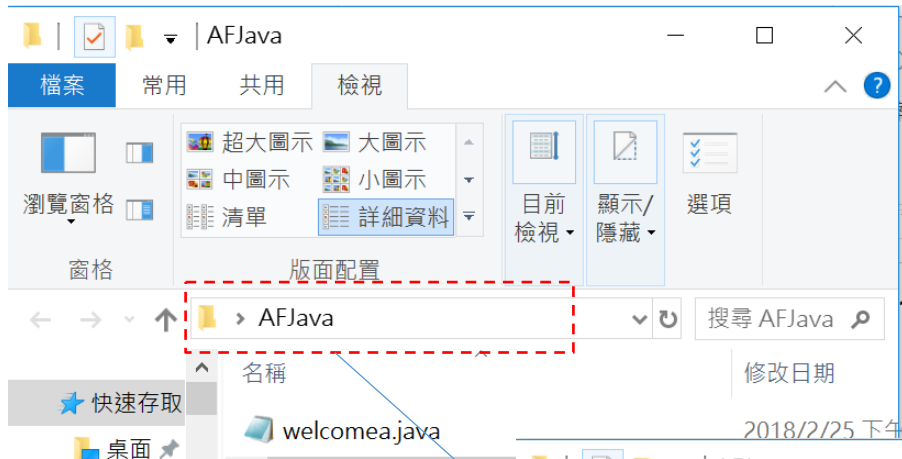
public class welcomea{
    public static void main(String args[]){
        System.out.println("Java is an excellent computer language.");
        System.out.println("Let's start to learn Java!");
    } //main()
} //class
```



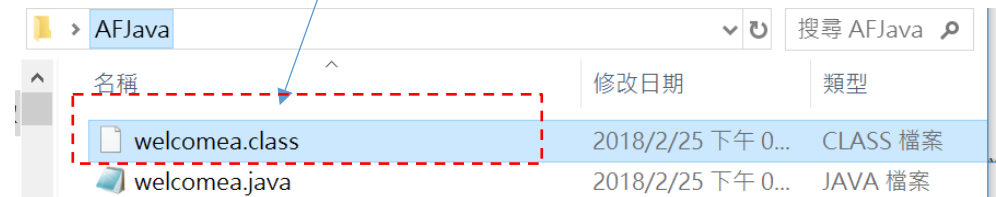
- 以類別名稱存檔，副檔名為java

# How to compile your first program?

- 進入cmd編譯程式:  
command line環境  
(MS DOS)



查看結果:  
產出.class



# How to run your first program?執行

- 錯誤

- Java welcomea.class

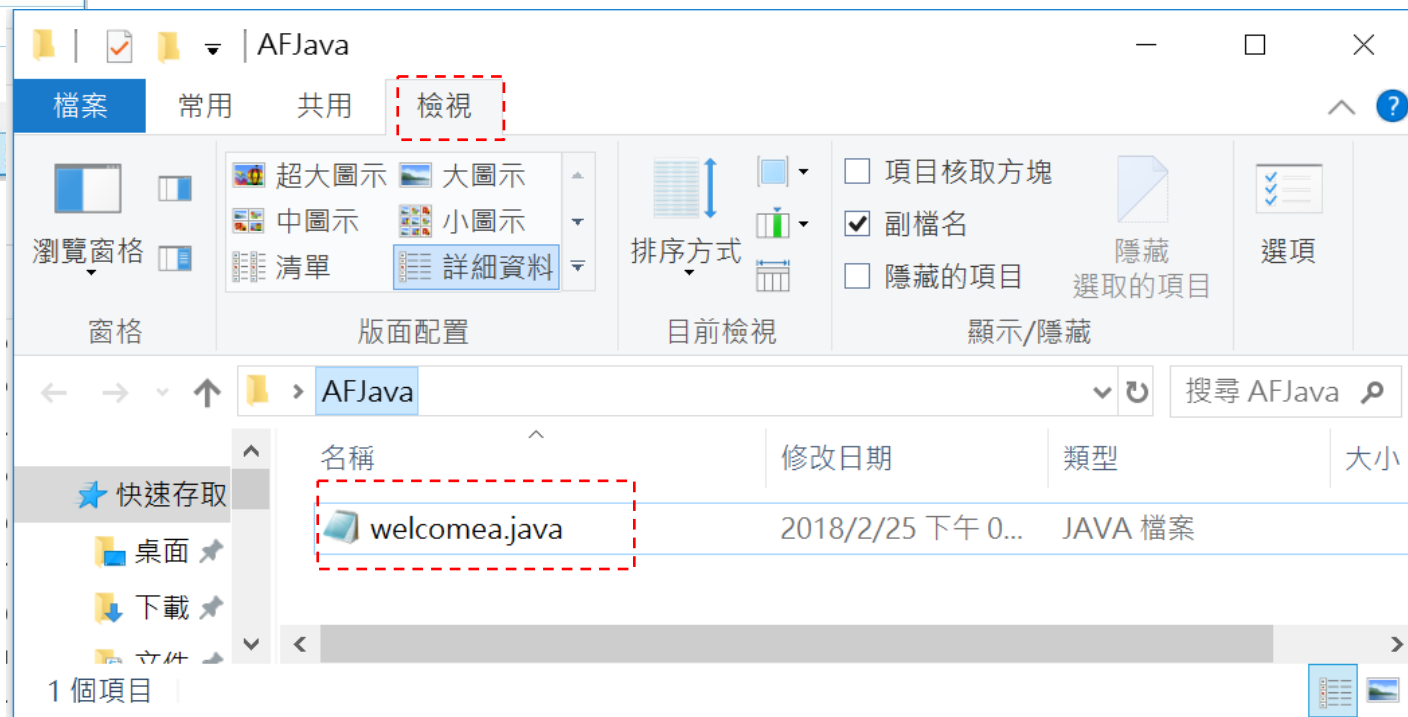
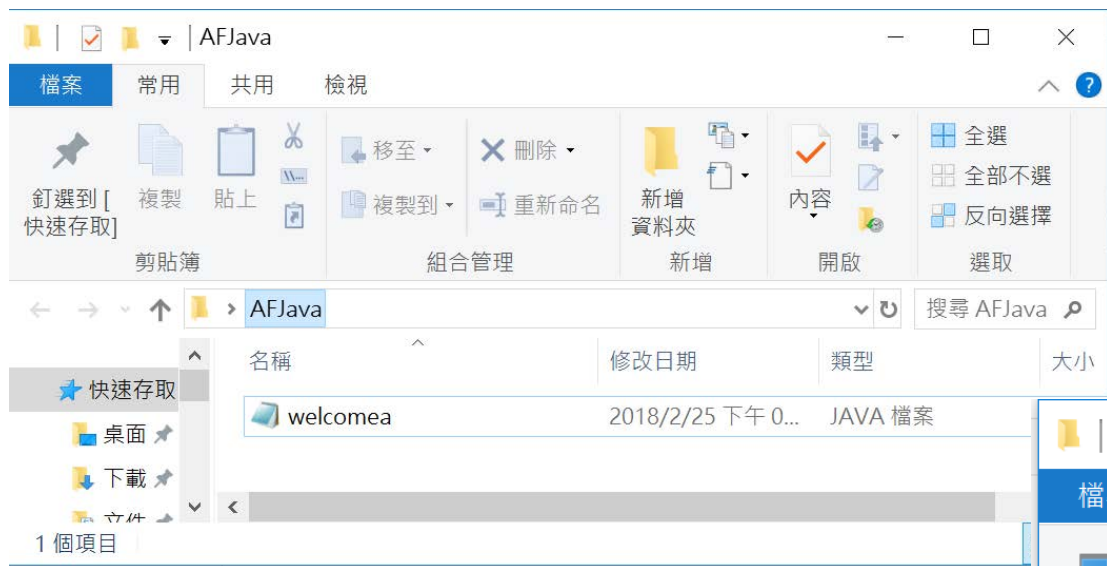
```
C:\Users\user\Desktop\AFJava>java welcomea.class  
錯誤：找不到或無法載入主要類別 welcomea.class
```

- 正確

- Java welcomea

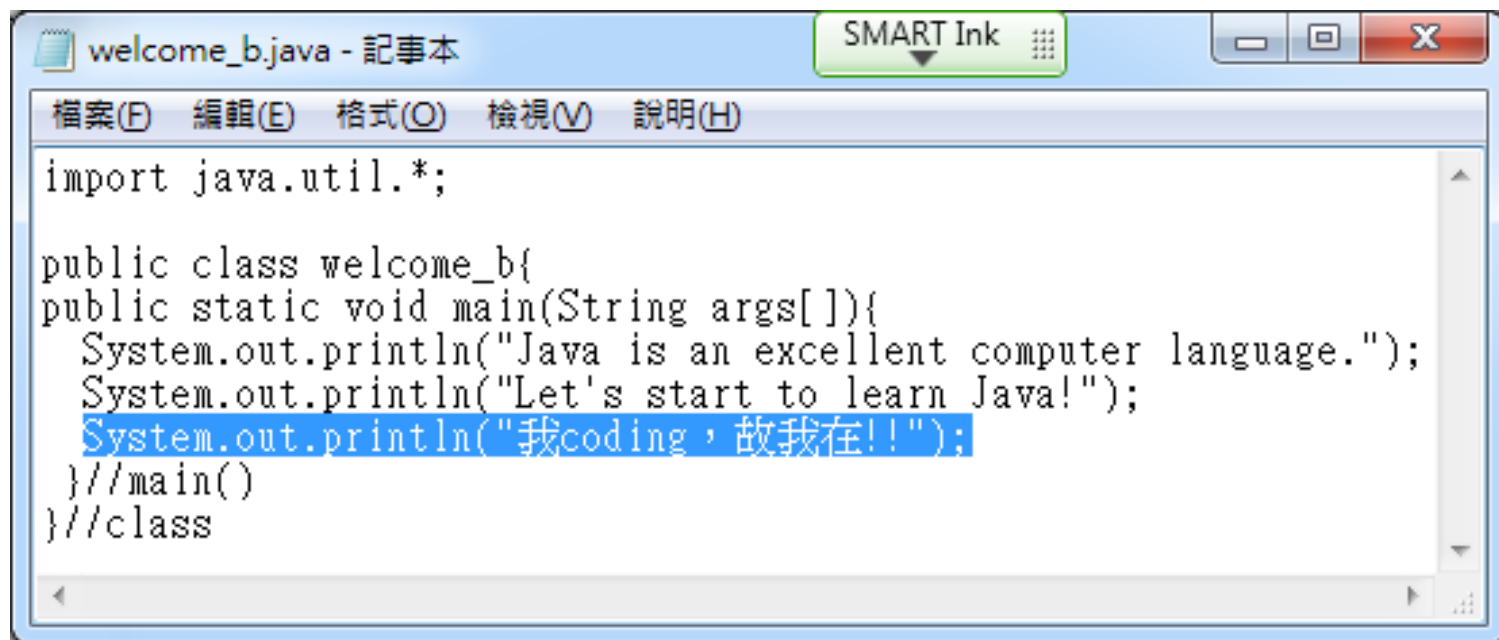
```
C:\Users\user\Desktop\AFJava>java welcomea  
Java is an excellent(很棒) computer language.  
Let's start to learn Java!
```

# 副檔名未出現,how to do?



# 開始練習!

- 仿作:EDIT, COMPILE, RUN
- 修改: 增加一行輸出，如



```
welcome_b.java - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
import java.util.*;

public class welcome_b{
public static void main(String args[]){
    System.out.println("Java is an excellent computer language.");
    System.out.println("Let's start to learn Java!");
    System.out.println("我coding，故我在!!");
} //main()
} //class
```

# 程式回顧

```
import java.util.*; //匯入函數庫
```

```
public class welcomea{
```

```
public static void main(String args[]){ //主程式名稱固定
```

```
//主程式body
```

```
}//main()結尾
```

```
}//class結尾
```

Class body

# 仔細研究程式

```
import java.util.*; //匯入函數庫
public class welcomea{
    public static void main(String args[]){
        System.out.println("Java is an excellent computer language."); //輸出
        System.out.println("Let's start to learn Java!");
    } //main()
} //class
```

# 程式概念

- Java是物件導向(OOP)語言，因此在設計都須將程式視為物件之類別(class)，給予類別名稱(class name)，且類別名稱一定是主檔案名稱(主檔名)，副檔名一定是java。
- 類別名稱如何命名?
  - 以英文字開頭，取有意義名稱，勿用中文
- Java程式由許多函數/方法(function/method)所組成，函數是由許多敘述或指令所組成。



# 程式概念

- CPU如何知道開始要執行哪一函數?CPU很笨!
  - 因此須規定開始執行的函數(entry point)，那就是main，main即為主函數。
- 大部分電腦都不是自然語言，因此會先規定一些字彙做為特殊功能，即保留字(reserved word)，這些字有特殊用途，不可任意使用
  - class, public, static, void
- 區分大小寫(case sensitive):各種名稱、保留字
- . :class/object存取符號，“的”
- “ ”: 字串常數
- // :註解(comment), 說明用，不是指令，編譯時會被忽略

# 習題2-1:編輯你的第一支java程式，完成java程式三部曲

- 為你的第一支java程式取名字，如AF\_first.java
  - 名字規則:英文字開頭、區分大小寫、第二字元後可用數字\_，但不可用+-\*/
- 輸出敘述至少四行，寫出第一支java程式感受，中英文不限
  - System.out.println(" !");
- 上傳作業(.java及執行畫面截圖放.docx)至酷課雲

```
import java.util.*; //匯入函數庫
public class AF_first {
    public static void main(String args[]){
        System.out.println(""); //輸出
        System.out.println("");
        System.out.println("");
        System.out.println("");
    } //main()
} //class
```

# Debug抓錯蟲

```
import java.util.*; /匯入函數庫
```

```
public class 9welcomex{
```

```
public static void Main(String args[]){
```

```
    system.Out.println("Java is an excellent computer language.");/輸出
```

```
    system.Out.println('Let's start to learn Java!')
```

```
}/main()
```

```
}/class
```

# Second program

```
import java.util.*;

public class welcome_c{

public static void main(String args[]){

    System.out.print("Please input your name:");

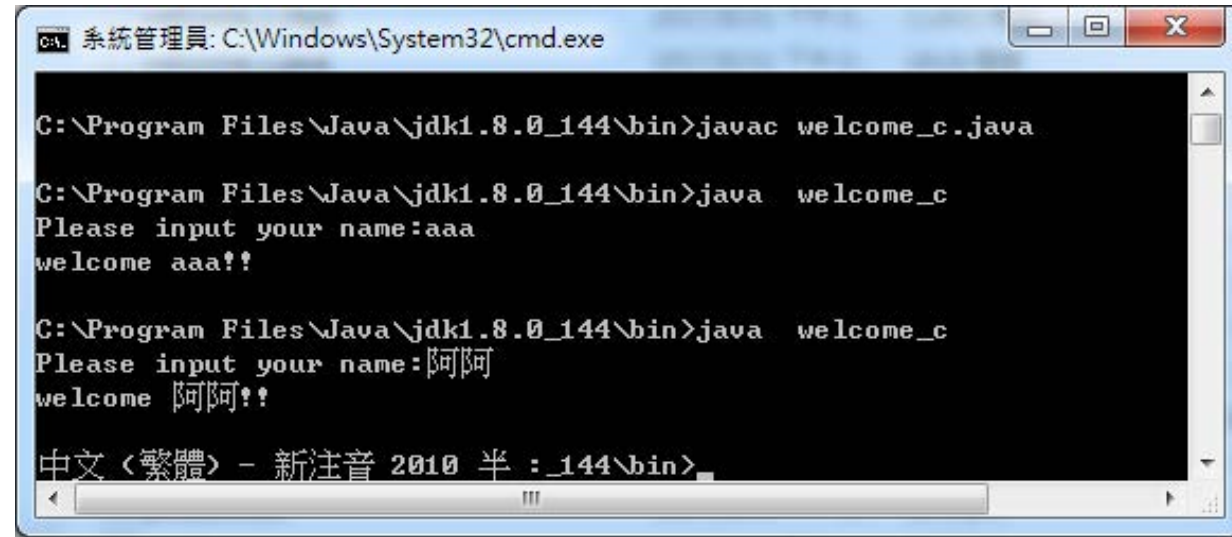
    Scanner sca = new Scanner(System.in);//產生Scanner物件, 提供鍵盤輸入

    String name = sca.next();

    System.out.println("welcome "+name+"!!");

} //main()

} //class
```



The screenshot shows a Windows command prompt window titled "系統管理員: C:\Windows\System32\cmd.exe". The window contains the following text:

```
C:\Program Files\Java\jdk1.8.0_144\bin>javac welcome_c.java

C:\Program Files\Java\jdk1.8.0_144\bin>java welcome_c
Please input your name:aaa
welcome aaa!!

C:\Program Files\Java\jdk1.8.0_144\bin>java welcome_c
Please input your name:阿阿
welcome 阿阿!!

中文 <繁體> - 新注音 2010 半 :_144\bin>
```

# Debug(偵錯)之由來

在二戰結束後，霍普（Grace Murray Hopper，1906-1992）在哈佛大學與同事開始建造第二代的電腦（Mark II）。夏日的某一天，Mark II忽然不動了。霍普打開電腦檢查，看到一隻蛾卡在裡面；於是就拿了一把鑷子把蟲給夾出來。從此電腦的除錯就叫做debug了。  
參考文獻：Darlene R. Stille, Extraordinary women scientists.



# 何謂偵(抓)錯(debug)?語法(syntax)·語意(semantic)錯誤? 何謂執行程式?

- 作業系統之**Loader**程式負責載入**RAM**,再交給**CPU**執行
- 為何需要**Library**函式庫?
  - 站在巨人肩膀,有些程式細節非常複雜如**Input/Output**、圖形之細部工作,因此必須提供系統函數提供設計者直接引用.
- 語法(**syntax**)錯誤?
  - 指的是句子文法上的錯誤,ie:我是學生 → 我學生是
  - **Compiler**可抓到語法錯誤
- 語意(**semantic**)錯誤?
  - 指的是句子意思上的錯誤,ie:我是學生 → 我不是學生
  - **Compiler**抓不到語意錯誤