

# Compiler Laboratory #1

NTUCSIE Compiler 2011 (03/07)

實驗一、三、四是系列實驗，針對的語言是 C 簡簡剪剪減減減。實驗一目的是實作其 scanner。

要判斷的 tokens 整理如下：

- operator / punctuation

token	token code	token	token code	token	token code	token	token code
+	43	<	60	(	40	,	44
-	45	>	62	)	41	;	59
*	42	<=	258	[	91	=	61
/	47	>=	259	]	93		
&&	256	==	260	{	123		
	257	!=	261	}	125		
!	33						

註：單一字元的 token，其 token code 即為其 ASCII。

- keyword

token	token code	token	token code	token	token code	token	token code
int	262	if	263	break	266	scan	268
		else	264	continue	267	print	269
		while	265				

- identifier

token	token code	attribute	說明
id	270	即其名稱	字母帶頭，後面跟字母、數字或底線

- literal

token	token code	attribute	說明
num	271	即其數值	僅需考慮十進位整數；0~999999
string	272	把\"與\\替代成\"與\\後， 不含前後雙引號的字串	雙引號括起來的字串； 字串內允許的字元為 printable ASCII 此外僅需考慮\"與\\兩個特殊字元

另外還需濾掉 0x20, 0x09, 0x0d, 0x0a 以及註解。註解是由 /\* 與往下第一個碰到的 \*/ 括起來的範圍，可以跨行。

輸入檔案：C 簡簡剪剪減減減程式原始檔，參見附檔之測試檔 lab\_1/\*.cmm。

輸出檔案：實作 scanner 後，另寫一支主程式（模擬 parser）重覆呼叫 scanner。每呼叫一次取得一個 token，隨後印出到輸出檔案。參見附檔之參考答案 lab\_1/\*.cmm.out。  
詳細規範如下：

- 輸出檔案名稱即輸入檔案名稱加上 .out。
- 一行一個 token。
- 沒有 attribute 的 token，僅須印出其 token code，前後不要有任何空白字元。
- 有 attribute 的 token，印出 token code 後，列印一個 tab，然後列印其 attribute。
- 針對 num，attribute 列印方式為 32 bits 的 16 進位表示法，英文字母小寫。如 255 的 attribute 列印 0x000000ff，不要用 0XFF。

開發工具：語言不拘，工具不拘，亦可使用任一 scanner generator (假如你無特殊喜好，建議用 flex)，只要寫入 Makefile 即可。宜考量到與實驗三、四的接軌。

執行平台：217 工作站實驗室之 linux $\alpha$ 。

執行環境：解壓縮附檔，即可得下列目錄結構。

```
lab_1/
├── b97902888/
│   ├── Makefile (請修正)
│   ├── scanner 程式原始檔 (待開發，附檔的 Fake.java 是玩假的)
│   ├── scanner 程式執行檔 (由 make 產生)
│   └── *.cmm.out (scanner 執行結果，由 make 產生)
├── lab_1.pdf (本說明檔案)
├── C簡簡剪剪減減減.pdf
├── Makefile (驗收時用，請勿修改)
├── *.cmm (測試檔，驗收時會增加)
└── *.cmm.out (參考答案)
```

執行步驟：

1. 於 lab\_1/b97902888/ 目錄底下開發程式。必要時可以建立子目錄。
2. 修改 lab\_1/b97902888/Makefile。  
必須搭配驗收用之 lab\_1/Makefile。搭配細節請見附檔兩個 Makefile 內之註解。  
步驟 3~6 是驗收步驟。
3. cd lab\_1
4. make clean STU=b97902888  
效果必須清掉 lab\_1/b97902888/ 底下之程式執行檔以及執行結果。
5. make judge STU=b97902888 PROG=tokenA.cmm  
效果包含：
  - (1) 產生程式執行檔。執行檔擺放位置不得超出目錄 lab\_1/b97902888/ 之範圍。
  - (2) 產生執行結果 lab\_1/b97902888/tokenA.cmm.out。請留意目錄以及檔名。
  - (3) diff 執行結果 lab\_1/b97902888/tokenA.cmm.out 與參考答案 lab\_1/tokenA.cmm.out。若沒有比對出任何差異，則通過。
6. make judge STU=b97902888 PROG=tokenB.cmm  
一次 make judge 餵入一份測試檔。  
附檔提供的測試檔僅是範例。驗收時，實驗要求還是以本文件的規範為準。建議應多測試自創之測試檔。
7. cd lab\_1/b97902888
8. make run PROG=自創測試檔  
效果為 5(1) 與 5(2)，不含 5(3) 的 diff。

結果繳交：請上傳一個 rar 或是 zip 檔到 ceiba 系統，結構必須為：

```
lab_1/
├── b97902888/
│   ├── Makefile
│   ├── scanner 程式原始檔
│   ├── 額外功能測試檔 (optional)
│   └── IWantToSay.txt (optional)
```

解壓縮後，第一層必須是 lab\_1。第二層必須是你的學號 (不要 888)，英文字母小寫。請參考附檔之結構。只需上傳程式原始檔 (不要 Fake.java) 以及 lab\_1/b97902888/Makefile，其他檔案不得上傳 (額外功能測試檔例外)。假如有話要交代助教，可以多加 IWantToSay.txt。

上傳後，建議自行下載一次，以確認繳交的是正確的版本。

額外功能：倘若你的成果有額外功能，請附上展現額外功能的測試檔，請補充 lab1/b97902888/Makefile，並請於 IWantToSay.txt 說明：(1) 額外功能為何 (2) 如何執行 make 以展現額外功能。

建議可以添加的額外功能 (但不侷限)：error detection, error recovery, ... 等。

評分原則：

1. 繳交截止時間為 3/26 週六中午 12:00。寬限時間為 1 小時。
2. 週六 13:00~週日 12:00 算遲交 1 天。週日 12:00~週一 12:00 算遲交 2 天。以下依此類推，除第 1 天外，均以 12:00 為分界點。
3. 遲交 1~7 日，每日扣得分 10%。第 8 日起，以 0 分計算。
4. 結果正確性與完整性佔 90%；部分通過則斟酌略減。
5. Program readability/maintainability 佔 10%。
6. 有額外功能則斟酌加分。