Simple Java － Scanner

B113040052 陳育霖

Lex 版本

使用版本：**flex 2.6.4**

作業平台

作業系統：**Ubuntu 22.04.4 LTS**

編譯器：**gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0**

工具鏈：flex、gcc、make

執行方式

1. **使用 make 指令自動編譯：**

   ```
   make
   ```

2. **會生成執行檔 demo**

   執行 demo 並輸入測試檔：

   ```
   ./demo < test.java
   ```

3. **結果會輸出每個 token 的：**

   類型（如：ID、float、operator）

   所在的行號

   該行的第幾個字元

   例如：Line: 1, 1st char: 1, "// print hello world" is a "comment".

4. **最後輸出 Symbol Table 中所有的識別字（Identifier）**

   例如：

   ```
   The symbol table contains:
   a
   ```

處理規格書上的方式

Token 的分類與判斷處理：

為符合規格書所提的分類需求，我將 Scanner 所辨識的 token 分為以下幾類：

Reserved Words（保留字）：以字串表的方式將所有 reserved word 儲存（如 int, float, if, else…等），直接在 Lex 中用條件語法判斷是否屬於保留字。

Identifiers（識別字）：用正規表示式[a-zA-Z_][a-zA-Z0-9_]*，在辨識到後檢查是否為保留字，是的話則分類為 reserved word，否則儲存至 symbol table。

整數（integer）：用 [0-9]+ 處理，額外加入負號支援（例如：-123）。

浮點數（float）：處理 1.0, 3.14, -0.5, 12.5e+3, -1.2E-5…等格式，使用多組 regex 結合，並加上狀態切換避免誤判。

String（字串常數）：正規表示式處理雙引號包住的內容，並支援 \" 內嵌的雙引號，例如："aa\"bb"。

Operators & Symbols（運算子與符號）：個別設立規則對應所有符號與運算子（如：+, ++, =, <=, !=, (,)…等），用 Lex 的固定字串處理。

Comments（註解）：

單行註解： //：掃描至該行結尾為止。

多行註解： **/* ... */**：使用 start condition 處理多行狀態，可處理註解內嵌註解結構，不跨越行未結束註解會提示錯誤。

錯誤處理與 recovery：

遇到非法字元時會輸出錯誤訊息，並跳過繼續處理後續 token。

例如：Line: 1, 1st char: 1, "1a" is a "error".

Symbol Table 的建立與管理：

hash = (hash * 33 + c) % table_size；c = 該名稱之字元。

create()：初始化 symbol table。

lookup(char* s)：檢查字串是否已存在，存在回傳 index，否則回傳 -1。

insert(char* s)：若字串不存在，加入 symbol table。

dump()：列印所有已儲存的識別字。

遇到的問題與解決方式

1. 遇到無內容的換行會一直被判定為 error

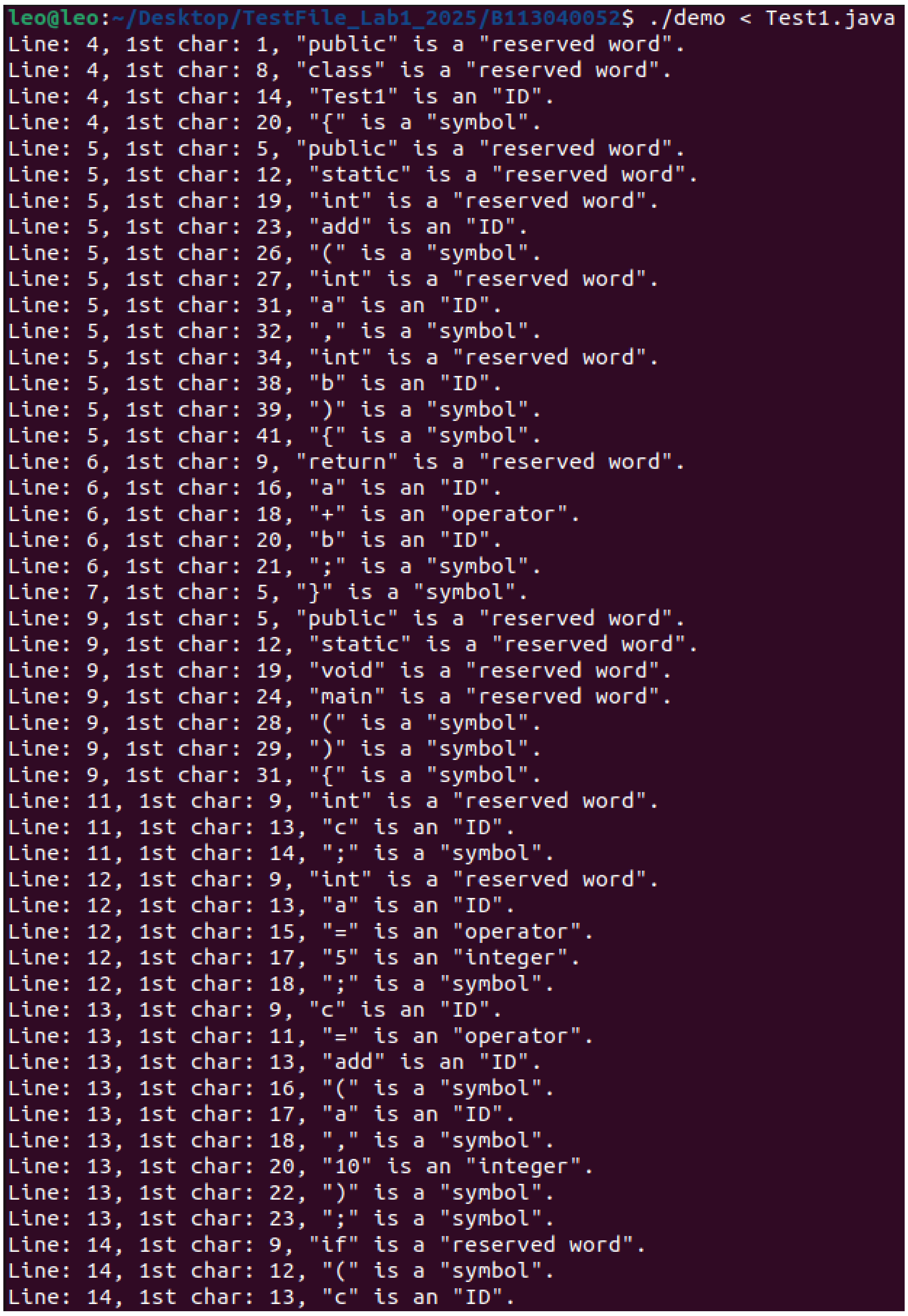
   解決方法：加上\r (Carriage Return) \v (垂直的 tab) \f (換頁) 的判斷

2. 字串會將開頭與結尾的雙引號"讀取並輸出

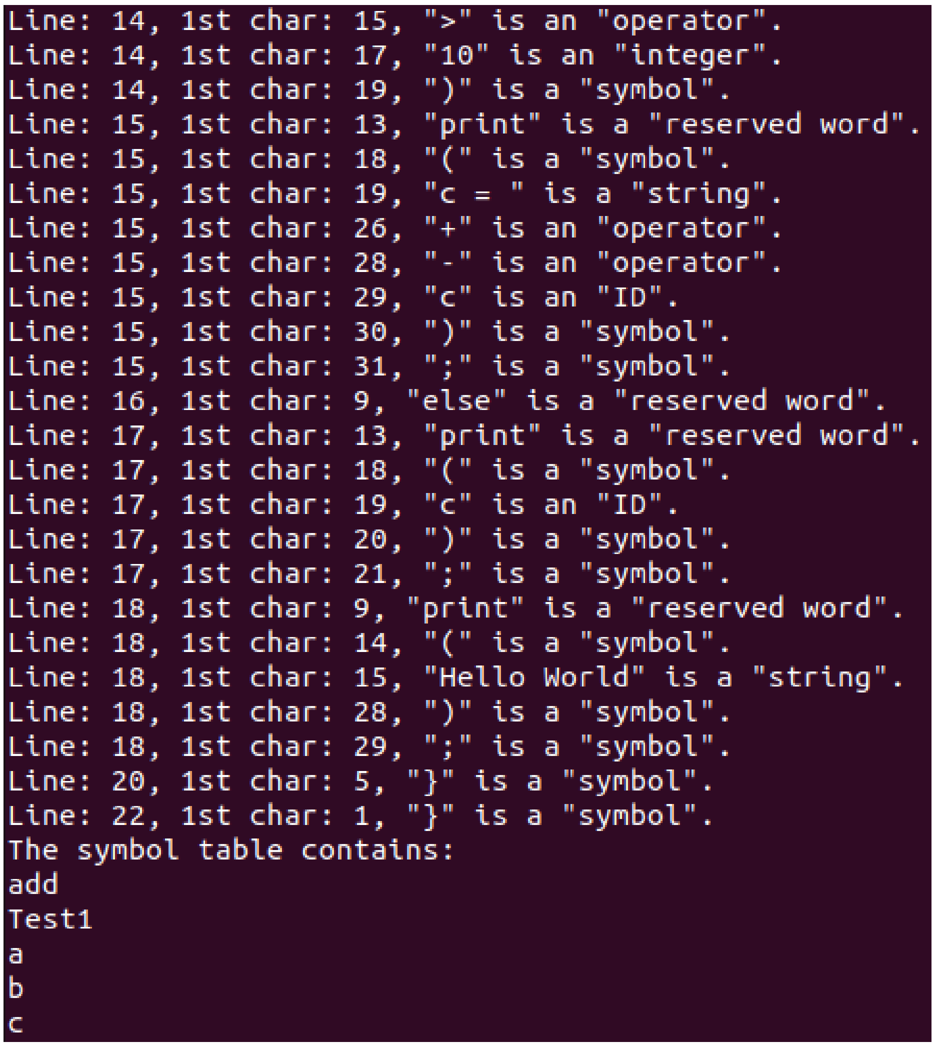
   解決方法：將所有字元往前移一格，並將倒數第二個字元改成\0

測試檔執行出來的結果

Test1.java

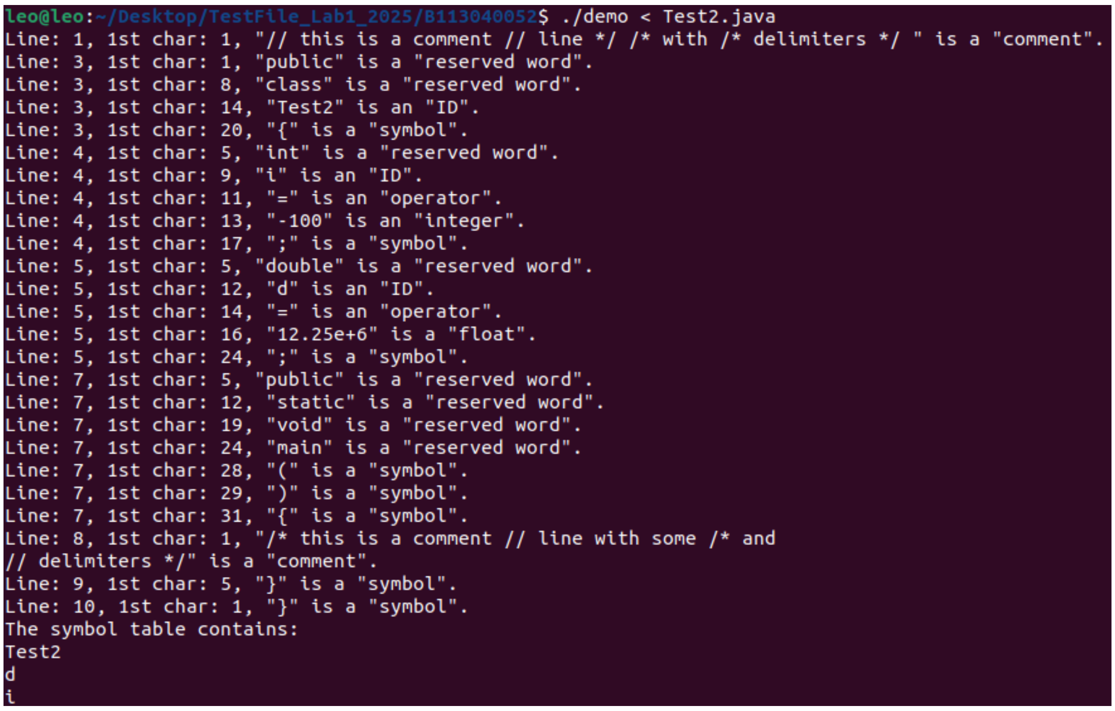


```
leo@leo:~/Desktop/TestFile_Lab1_2025/B113040052$ ./demo < Test1.java
Line: 4, 1st char: 1, "public" is a "reserved word".
Line: 4, 1st char: 8, "class" is a "reserved word".
Line: 4, 1st char: 14, "Test1" is an "ID".
Line: 4, 1st char: 20, "{" is a "symbol".
Line: 5, 1st char: 5, "public" is a "reserved word".
Line: 5, 1st char: 12, "static" is a "reserved word".
Line: 5, 1st char: 19, "int" is a "reserved word".
Line: 5, 1st char: 23, "add" is an "ID".
Line: 5, 1st char: 26, "(" is a "symbol".
Line: 5, 1st char: 27, "int" is a "reserved word".
Line: 5, 1st char: 31, "a" is an "ID".
Line: 5, 1st char: 32, "," is a "symbol".
Line: 5, 1st char: 34, "int" is a "reserved word".
Line: 5, 1st char: 38, "b" is an "ID".
Line: 5, 1st char: 39, ")" is a "symbol".
Line: 5, 1st char: 41, "{" is a "symbol".
Line: 6, 1st char: 9, "return" is a "reserved word".
Line: 6, 1st char: 16, "a" is an "ID".
Line: 6, 1st char: 18, "+" is an "operator".
Line: 6, 1st char: 20, "b" is an "ID".
Line: 6, 1st char: 21, ";" is a "symbol".
Line: 7, 1st char: 5, "}" is a "symbol".
Line: 9, 1st char: 5, "public" is a "reserved word".
Line: 9, 1st char: 12, "static" is a "reserved word".
Line: 9, 1st char: 19, "void" is a "reserved word".
Line: 9, 1st char: 24, "main" is a "reserved word".
Line: 9, 1st char: 28, "(" is a "symbol".
Line: 9, 1st char: 29, ")" is a "symbol".
Line: 9, 1st char: 31, "{" is a "symbol".
Line: 11, 1st char: 9, "int" is a "reserved word".
Line: 11, 1st char: 13, "c" is an "ID".
Line: 11, 1st char: 14, ";" is a "symbol".
Line: 12, 1st char: 9, "int" is a "reserved word".
Line: 12, 1st char: 13, "a" is an "ID".
Line: 12, 1st char: 15, "=" is an "operator".
Line: 12, 1st char: 17, "5" is an "integer".
Line: 12, 1st char: 18, ";" is a "symbol".
Line: 13, 1st char: 9, "c" is an "ID".
Line: 13, 1st char: 11, "=" is an "operator".
Line: 13, 1st char: 13, "add" is an "ID".
Line: 13, 1st char: 16, "(" is a "symbol".
Line: 13, 1st char: 17, "a" is an "ID".
Line: 13, 1st char: 18, "," is a "symbol".
Line: 13, 1st char: 20, "10" is an "integer".
Line: 13, 1st char: 22, ")" is a "symbol".
Line: 13, 1st char: 23, ";" is a "symbol".
Line: 14, 1st char: 9, "if" is a "reserved word".
Line: 14, 1st char: 12, "(" is a "symbol".
Line: 14, 1st char: 13, "c" is an "ID".
```



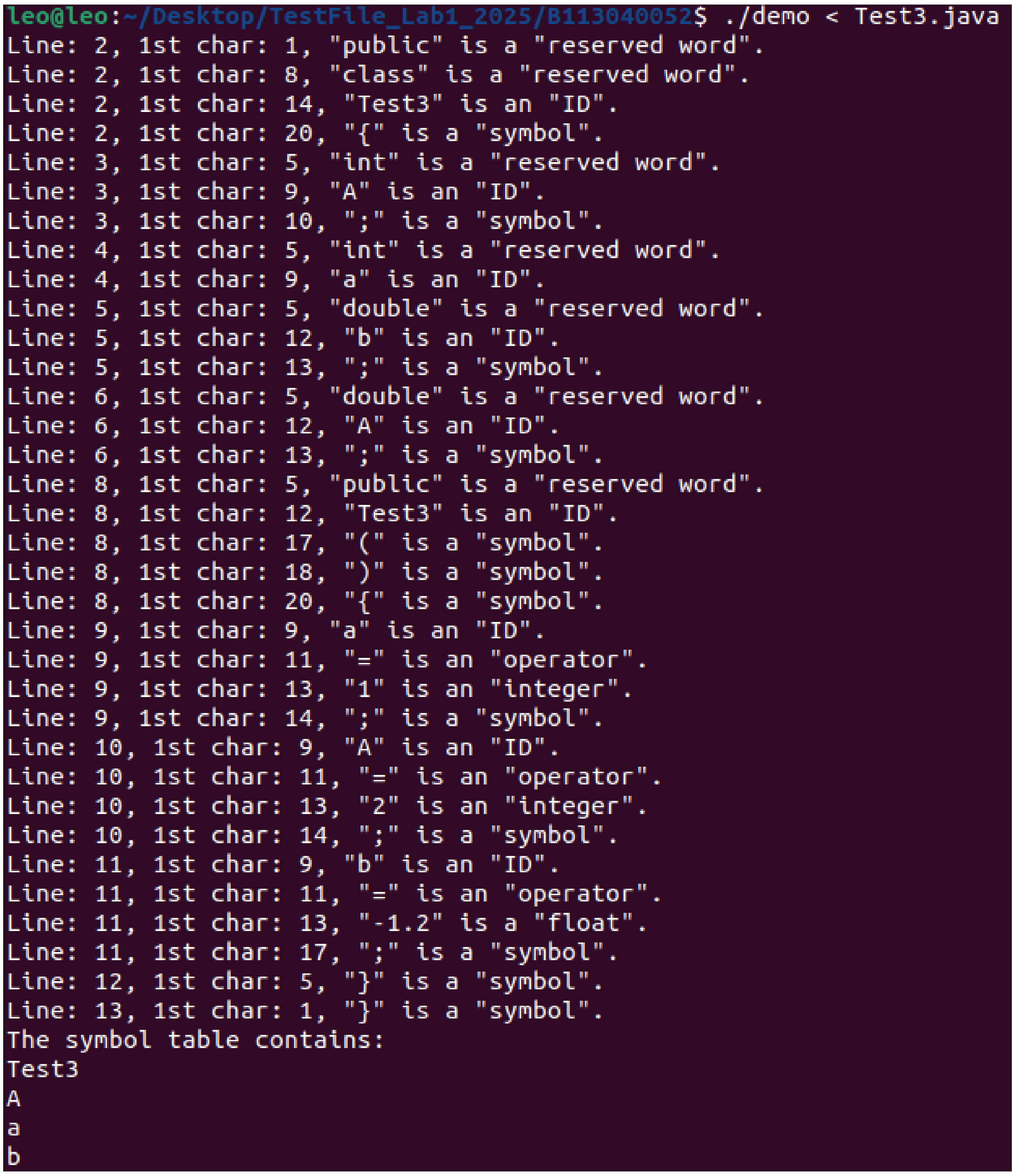
```
Line: 14, 1st char: 15, ">" is an "operator".
Line: 14, 1st char: 17, "10" is an "integer".
Line: 14, 1st char: 19, ")" is a "symbol".
Line: 15, 1st char: 13, "print" is a "reserved word".
Line: 15, 1st char: 18, "(" is a "symbol".
Line: 15, 1st char: 19, "c = " is a "string".
Line: 15, 1st char: 26, "+" is an "operator".
Line: 15, 1st char: 28, "-" is an "operator".
Line: 15, 1st char: 29, "c" is an "ID".
Line: 15, 1st char: 30, ")" is a "symbol".
Line: 15, 1st char: 31, ";" is a "symbol".
Line: 16, 1st char: 9, "else" is a "reserved word".
Line: 17, 1st char: 13, "print" is a "reserved word".
Line: 17, 1st char: 18, "(" is a "symbol".
Line: 17, 1st char: 19, "c" is an "ID".
Line: 17, 1st char: 20, ")" is a "symbol".
Line: 17, 1st char: 21, ";" is a "symbol".
Line: 18, 1st char: 9, "print" is a "reserved word".
Line: 18, 1st char: 14, "(" is a "symbol".
Line: 18, 1st char: 15, "Hello World" is a "string".
Line: 18, 1st char: 28, ")" is a "symbol".
Line: 18, 1st char: 29, ";" is a "symbol".
Line: 20, 1st char: 5, "}" is a "symbol".
Line: 22, 1st char: 1, "}" is a "symbol".
The symbol table contains:
add
Test1
a
b
c
```

Test2.java



```
leo@leo:~/Desktop/TestFile_Lab1_2025/B113040052$ ./demo < Test2.java
Line: 1, 1st char: 1, "// this is a comment // line */ /* with /* delimiters */ " is a "comment".
Line: 3, 1st char: 1, "public" is a "reserved word".
Line: 3, 1st char: 8, "class" is a "reserved word".
Line: 3, 1st char: 14, "Test2" is an "ID".
Line: 3, 1st char: 20, "{" is a "symbol".
Line: 4, 1st char: 5, "int" is a "reserved word".
Line: 4, 1st char: 9, "i" is an "ID".
Line: 4, 1st char: 11, "=" is an "operator".
Line: 4, 1st char: 13, "-100" is an "integer".
Line: 4, 1st char: 17, ";" is a "symbol".
Line: 5, 1st char: 5, "double" is a "reserved word".
Line: 5, 1st char: 12, "d" is an "ID".
Line: 5, 1st char: 14, "=" is an "operator".
Line: 5, 1st char: 16, "12.25e+6" is a "float".
Line: 5, 1st char: 24, ";" is a "symbol".
Line: 7, 1st char: 5, "public" is a "reserved word".
Line: 7, 1st char: 12, "static" is a "reserved word".
Line: 7, 1st char: 19, "void" is a "reserved word".
Line: 7, 1st char: 24, "main" is a "reserved word".
Line: 7, 1st char: 28, "(" is a "symbol".
Line: 7, 1st char: 29, ")" is a "symbol".
Line: 7, 1st char: 31, "{" is a "symbol".
Line: 8, 1st char: 1, "/* this is a comment // line with some /* and
// delimiters */" is a "comment".
Line: 9, 1st char: 5, "}" is a "symbol".
Line: 10, 1st char: 1, "}" is a "symbol".
The symbol table contains:
Test2
d
i
```

Test3.java



```
leo@leo:~/Desktop/TestFile_Lab1_2025/B113040052$ ./demo < Test3.java
Line: 2, 1st char: 1, "public" is a "reserved word".
Line: 2, 1st char: 8, "class" is a "reserved word".
Line: 2, 1st char: 14, "Test3" is an "ID".
Line: 2, 1st char: 20, "{" is a "symbol".
Line: 3, 1st char: 5, "int" is a "reserved word".
Line: 3, 1st char: 9, "A" is an "ID".
Line: 3, 1st char: 10, ";" is a "symbol".
Line: 4, 1st char: 5, "int" is a "reserved word".
Line: 4, 1st char: 9, "a" is an "ID".
Line: 5, 1st char: 5, "double" is a "reserved word".
Line: 5, 1st char: 12, "b" is an "ID".
Line: 5, 1st char: 13, ";" is a "symbol".
Line: 6, 1st char: 5, "double" is a "reserved word".
Line: 6, 1st char: 12, "A" is an "ID".
Line: 6, 1st char: 13, ";" is a "symbol".
Line: 8, 1st char: 5, "public" is a "reserved word".
Line: 8, 1st char: 12, "Test3" is an "ID".
Line: 8, 1st char: 17, "(" is a "symbol".
Line: 8, 1st char: 18, ")" is a "symbol".
Line: 8, 1st char: 20, "{" is a "symbol".
Line: 9, 1st char: 9, "a" is an "ID".
Line: 9, 1st char: 11, "=" is an "operator".
Line: 9, 1st char: 13, "1" is an "integer".
Line: 9, 1st char: 14, ";" is a "symbol".
Line: 10, 1st char: 9, "A" is an "ID".
Line: 10, 1st char: 11, "=" is an "operator".
Line: 10, 1st char: 13, "2" is an "integer".
Line: 10, 1st char: 14, ";" is a "symbol".
Line: 11, 1st char: 9, "b" is an "ID".
Line: 11, 1st char: 11, "=" is an "operator".
Line: 11, 1st char: 13, "-1.2" is a "float".
Line: 11, 1st char: 17, ";" is a "symbol".
Line: 12, 1st char: 5, "}" is a "symbol".
Line: 13, 1st char: 1, "}" is a "symbol".
The symbol table contains:
Test3
A
a
b
```