OURC_Project 1

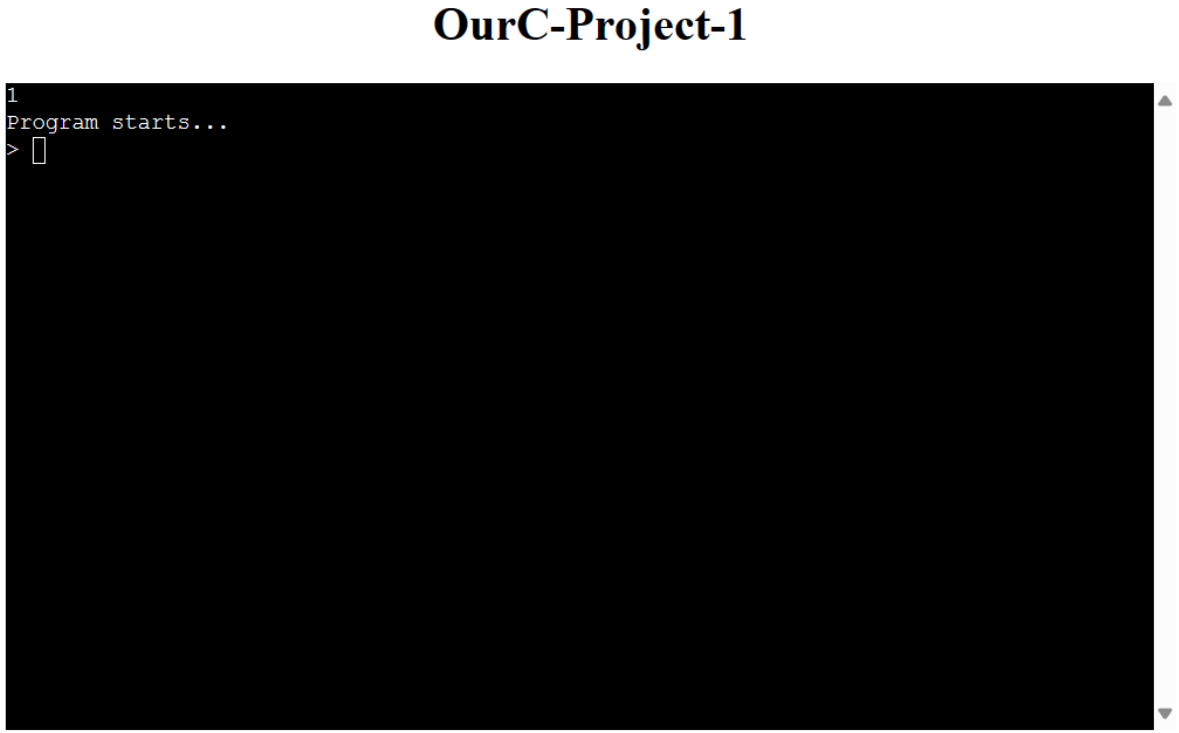
DESIGN : GAXZIPTGO

outline

- 測試平台
- Project 1
  - GetToken
  - Syntax Analysis
  - Process
- About PAL
  - 一些盡量避免的寫法
  - 如何”模仿”PAL輸出

測試平台

- URL: https://pl.lab214b.uk:10000/



GetToken(Project 1)

- normal: 將token分類成定義好的type

Type	Token
IDENT	a , ab_123, abc123_
NUM	123, 123.234, .123, 123. , 07
SIGN	+, -
QUIT	quit
others	:=, <>, >, <, =, <=, >=, (,) *, /, ;

GetToken(Project 1)

- 容易有的問題:

IDENT: 一個letter開頭 後面銜接底線 letter || '_' || digital

ex: abc (O), abc_(O), abc_123(O), abc_123_(O), _abc(X)

NUM: 可以是 int || float

一個digital或' . ' 開頭, 一直往後讀直到不是digital

ex: 123, 123.234, .123, 123. , 07

其他符號: 後面只能出現已經定義好的char

GetToken(Project 1)

- 容易有的問題:

  abc_123.123 := abc_123, .123

  123.123.123 := 123.123, .123

  abc_<>.123abc := abc_, <>, .123, abc

  abc_<>.123_abc := abc_, <>, .123, error

GetToken(Project 1)

- 關於錯誤(unrecognized char):
  - 1.遇到無法辨識的char
  - 2.無法分類的token
  - Ex : Abc$234 : :=’ $ ’ unrecognized

    _123 : :=’ _ ’ unrecognized

    : : :=’ : ’ unrecognized (但:=是可以的)

    . : :=‘ . ’ unrecognized

Syntax Analysis (Project 1)

```
<Command> ::= IDENT ( ':=' <ArithExp> | <IDlessArithExpOrBexp> ) ';'
            | <NOT_IDStartArithExpOrBexp> ';'
            | QUIT

<IDlessArithExpOrBexp>  ::= {   '+' <Term>   | '-' <Term>
                              | '*' <Factor> | '/' <Factor>
                            }
                            [ <BooleanOperator> <ArithExp> ]

<BooleanOprator>        ::= '=' | '<>' | '>' | '<' | '>=' | '<='

<NOT_ID_StartArithExpOrBexp>
                        ::= <NOT_ID_StartArithExp>
                            [ <BooleanOperator> <ArithExp> ]

<NOT_ID_StartArithExp>  ::= <NOT_ID_StartTerm> { '+' <Term> | '-' <Term> }
<NOT_ID_StartTerm>      ::= <NOT_ID_StartFactor> { '*' <Factor> | '/' <Factor> }
<NOT_ID_StartFactor>    ::= [ SIGN ] NUM | '(' <ArithExp> ')'
<ArithExp>              ::= <Term> { '+' <Term> | '-' <Term> }
<Term>                  ::= <Factor> { '*' <Factor> | '/' <Factor> }
<Factor>                ::= IDENT | [ SIGN ] NUM | '(' <ArithExp> ')'
```

Syntax Analysis (Project 1)

- 關於符號
  - () : 括弧內的一定要出現一次
  - [] : 括弧內的可以出現零次或一次
  - {} : 括弧內的可以出現零次或無數次
  - : := :該句型的定義

Syntax Analysis (Project 1)

- 關於錯誤 (unexpect token)

<Factor> ::= **IDENT** | [**SIGN**] **NUM** | '(' <ArithExp> ')'

- 例: (, !<ArithExp>,) ->(X)
- SIGN, !NUM ->(X)

<NOT_ID_StartArithExpOrBexp> ::= <NOT_ID_StartArithExp> [<BooleanOperator> <ArithExp>]

- !<NOT_ID_StartArithExp> ->(X)
- <NOT_ID_StartArithExp> !<BooleanOperator> ->(O)
- <NOT_ID_StartArithExp> <BooleanOperator> !<ArithExp> ->(X)

Syntax Analysis (Project 1)

- 關於錯誤 (undefined identifier)
- 規則:每讀到identifier 就應確認一次
- 若identifier開頭且下一個接 := 則不須處理 否則須確認是否定義
- 若identifier不在開頭 必須檢查!

Syntax Analysis (Project 1)

- 關於錯誤 (undefined identifier)

```
Undefined identifier : 'b'
> a := 4 ;
4
> b
+
Undefined identifier : 'b'
> a + b := 4 ;
Undefined identifier : 'b'
```

Process (Project 1)

- 關於輸入 輸出
- 以分號為結束, 輸出該段的結果

```
Program starts...
> 5;
5
> a:=3;
3
> a
;
3
> a+3;
6
> b:=4;
4
> a+b;
7
```

Process (Project 1)

- 關於define
- 將:=右邊運算後的輸出定義至左邊的 ID

```
Program starts...
> a := 4 ;
4
> b := a+ 4 ;
8
> c := a+ 4.0 ;
8.000
> b+c;
16.000
```

Process (Project 1)

- 關於NUM
- 整數+整數 = 整數, 整數+小數 = 小數, 小數+小數 = 小數

```
> 5+3 ;
8
> 5+3.0 ;
8.000
> 5.0 + 3.0 ;
8.000
> 5 / 3 ;
1
> 5 / 3.0 ;
1.667
> 5.0 / 3.0 ;
1.667
```

Process (Project 1)

- 關於BooleanOperator
  - = := 等於,
  - <> : := 不等於,
  - > : := 大於,
  - < : := 小於,
  - <= : := 小於等於,
  - >= : := 大於等於
  - 若相符 輸出true, 反之輸出false

Process (Project 1)

- 關於BooleanOperator
  - 比較的值以輸出的結果為主!

```
> a := 5/3 ;
1
> a <> 1 ;
false
> a = 1 ;
true
> a = 1.667 ;
false
```

About PAL

- PAL 是一套老系統 請不要用 ”所有” C++ 11特有的語法

- 所有參數都要做初始化
- 參考:-=神秘之旅=- 閱讀看板文章 (cycu.edu.tw)

- 盡量用struct 而不用class
- 若需要使用類似void function(char* &ch) 這種寫法 請先typedef char* something
- void function(something &ch)

- 關於vector的使用教學 (vector內不要放class !! 盡量用struct)
- 參考:-=神秘之旅=- 閱讀看板文章 (cycu.edu.tw)

About PAL

- 老大 : Vector 不要放在stack 放在heap裡
- Do not use this: vector< Token > uTempTokenList;
- Do this instead: vector< Token > * uTempTokenList;
- 參考 : -=神秘之旅=- 閱讀看板文章 (cycu.edu.tw)

About PAL

- 如何模仿PAL輸入:
- Windows cmd: type testfile.txt | yourProgram.exe
- Linux & MAC terminal: cat testfile | yourProgram

- 如果想要輸出成文件 可以如下
- Windows cmd:
- type testfile.txt | yourProgram.exe > outputfile.txt
- Linux & MAC terminal:
- cat testfile.txt | yourProgram > outputfile.txt