

Semantic Computing

語意計算

-PeterWolf

Syntax Error

Python

```
In [1]: print("Hello, World!"  
        Cell In[1], line 1  
          print("Hello, World!"  
              ^  
SyntaxError: incomplete input
```

Java

```
*FavouriteToy.java  
public class FavouriteToy {  
    // Since I have only one class,  
    // I must include the "public static main()" method in this class.  
    public static void main(String[] args) {  
        // Make an object to hang my "printToy()" method on.  
        Syntax error, insert ";" to complete BlockStatements  
        FavouriteToy favToy = new FavouriteToy();  
        // Check the number of arguments received as input.  
        // Default to "purple koalas" if number is not 2.  
        if (args.length == 2) {  
            favToy.printToy(args[0], args[1]);  
        } else {  
            favToy.printToy("purple", "koalas");  
        }  
    }  
}
```

C

```
C:\Users\Rashi\Desktop>gcc syntaxerror.c -o syntaxerror.exe  
syntaxerror.c: In function 'main':  
syntaxerror.c:9:3: error: expected declaration specifiers before 'printf'  
   9 |     printf("Geeks for Geeks");  
     |     ^~~~~~  
syntaxerror.c:10:3: error: expected declaration specifiers before 'return'  
  10 |     return 0;  
     |     ^~~~~~  
syntaxerror.c:11:1: error: expected declaration specifiers before '}' token  
  11 | }  
     | ^  
syntaxerror.c:12: error: expected '{' at end of input
```

C++

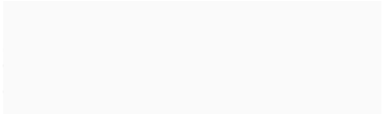
```
82 // Note CPU-Timer2 is reserved for TI use.  
83 Syntax error void INT14_ISR(void) // CPU-Timer2  
84 {  
85     // Insert ISR Code here  
86  
87     int16 i;  
88     for(i=1;i<10;i++);  
89  
90 //WGO   GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1;  
91  
92 }  
93
```

What about semantic error?

- | | | |
|---|----------------|---|
| 1 | Syntax error | Pattern issues |
| 2 | Run-time error | Environment issues |
| 3 | Linker error | External module / Internal Memory Object issues |
| 4 | Logical error | Wrong outputs |
| 5 | Semantic error | Unexpected outputs |

Syntax	Semantic	Logical
Detected by the compiler	May be detected by compiler, linker, run-time	May be detected at run-time

Normally Syntax errors are detected by the compiler. Semantic errors **may be** detected by compiler, linker or semantic errors can detect at run-time. When concern about the Logic errors, we can identify logical errors **may be** detected at run-time. But sometimes these errors are not detected at all

Syntax	Semantic	Logical
Syntax error IF(avg>70)THEN grade= "fail" ELSE grade= "pass" END IF	IF(avg>70)THEN READ "PASS" ELSE DISPLAY "FAIL" END IF Total=m1+m2+m3 Avg=(m1*m2*m3)/3;	Logical error WHIL(x<=10) IF(avg>=70)THE total=m1+m2+m3 

- (3) a. Floyd is tall. (vague)
 b. Floyd is six feet tall. (potentially imprecise)

- (8) a. Floyd is taller than Clyde.
 b. $\llbracket \text{Floyd is tall.} \rrbracket > \llbracket \text{Clyde is tall.} \rrbracket$

- (9) a. #Floyd is taller than this is a ferret.
 b. #Floyd is taller than Clyde is unpleasant.

很熱
* 很溫

- (20) a. $\llbracket \text{very} \rrbracket^c = \lambda f_{\langle c, et \rangle} \lambda x . f(c')(x)$

where c' is identical to c except that the comparison class
in c' is $\{y : f(c)(y)\}$

$$(23) \quad \llbracket \text{tall} \rrbracket = \lambda d \lambda x . \mathbf{tall}(d)(x)$$

$$(26) \quad \text{a. } \llbracket \text{six feet tall} \rrbracket = \llbracket \text{tall} \rrbracket (\llbracket \text{six feet} \rrbracket) = \lambda x . \mathbf{tall}(\mathbf{6-feet})(x)$$

$$\text{b. } \llbracket \text{Floyd is six feet tall} \rrbracket = \llbracket \text{is six feet tall} \rrbracket (\llbracket \text{Floyd} \rrbracket) \\ = \mathbf{tall}(\mathbf{6-feet})(\mathbf{Floyd})$$

$$(28) \quad \llbracket \text{Floyd is POS tall} \rrbracket^c = \exists d [d > \mathbf{standard}_c(\mathbf{tall}) \wedge \mathbf{tall}(d)(\mathbf{Floyd})]$$

$$(31) \quad \llbracket \text{POS} \rrbracket^c = \lambda G_{\langle d, et \rangle} \lambda x . \exists d [d > \mathbf{standard}_c(G) \wedge G(d)(x)]$$

$$(32) \quad \text{a. } \llbracket \text{POS tall} \rrbracket^c = \llbracket \text{POS} \rrbracket^c (\llbracket \text{tall} \rrbracket^c) \\ = \lambda x . \exists d [d > \mathbf{standard}_c(\mathbf{tall}) \wedge \mathbf{tall}(d)(x)]$$

$$\text{b. } \llbracket \text{Floyd is POS tall} \rrbracket^c = \llbracket \text{POS tall} \rrbracket^c (\llbracket \text{Floyd} \rrbracket^c) \\ = \exists d [d > \mathbf{standard}_c(\mathbf{tall}) \wedge \mathbf{tall}(d)(\mathbf{Floyd})]$$

- (33)
- a. $\llbracket \text{POS} \rrbracket^c = \lambda G_{\langle d, et \rangle} \lambda P_{\langle e, t \rangle} \lambda x . \exists d [G(x)(d) \wedge d !> \mathbf{norm}(G)(P)]$
 - b. $\llbracket \text{POS } tall \rrbracket^c = \llbracket \text{POS} \rrbracket^c (\llbracket tall \rrbracket^c)$
 $= \lambda P_{\langle e, t \rangle} \lambda x . \exists d [\mathbf{tall}(d)(x) \wedge d !> \mathbf{norm}(\mathbf{tall})(P)]$
 - c. $\llbracket \text{POS } tall \text{ for a basketball player} \rrbracket^c$
 $= \llbracket \text{POS } tall \rrbracket^c (\llbracket \text{for a basketball player} \rrbracket^c)$
 $= \llbracket \text{POS } tall \rrbracket^c (\mathbf{basketball-player})$
 $= \lambda x . \exists d \left[\mathbf{tall}(d)(x) \wedge d !> \mathbf{norm}(\mathbf{tall})(\mathbf{basketball-player}) \right]$

(82) DIMENSIONAL

- a. #Clyde is slightly tall.
- b. #This board is slightly long.

(83) NON-DIMENSIONAL

- a. Clyde is slightly stupid.
- b. Clyde is slightly lazy.

(84) Clyde is $\left\{ \begin{array}{l} \text{really} \\ \text{a little} \\ \text{shockingly} \end{array} \right\} \left\{ \begin{array}{l} \text{stupid} \\ \text{lazy} \\ \text{ugly} \end{array} \right\}.$