

1.

**Strong typing:**

指在程式執行前及執行時，所有的 type error 都可被發現，java 即為 strong typing。

**Inheritance:**

是指 Sub Class(子類別)繼承 Super Class(父類別)後,就會自動取得父類別特性。

**Smalltalk:**

為歷史上第二個物件導向的程式語言，和第一個真正的 IDE 環境；它不同於一般的程式語言，包含三種特色: 物件導向程式語言、程式設計環境與應用開發環境。

**Lifetime:**

指變數被配置記憶體空間的期間。

**Coercion:**

指將某一型態的值強制轉換成為另一種型態之值；分為:

Widening: 某一型態的值轉換為另一型態,而值不會改變者稱之。

Narrowing: 將某一型態的值轉換為另一型態,而值會改變者。

**Operator associativity rule:**

當具有相同優先度的運算子同時出現時，決定運算子所表示的邏輯操作是由左到右還是由右到左進行(後者為指數運算)。

**Garbage collection:**

收集記憶體的方法，如 java 中，若沒有足夠的記憶體空間時，garbage collection 會將散落的記憶體收集再利用，但缺點是需要花費時間。

2. (PS. BNF grammar 不只一種)

$\langle S \rangle \rightarrow \langle a \rangle$

$\langle a \rangle \rightarrow \langle a \rangle \langle a \rangle \mid \langle i \rangle$

$\langle i \rangle \rightarrow 0 \mid 1$

3.

指出 grammar 錯誤的地方:  $\langle e \rangle \rightarrow \langle t \rangle + \langle e \rangle \mid \langle e \rangle * \langle t \rangle \mid (\langle e \rangle) \mid \langle t \rangle$  ;

並畫出 parse tree、導出 leftmost derivation

4.

此 grammar 不是 ambiguous，並說明原因。

5.

{P:  $x \geq 100$ }

6.

Static:  $a = 1$  ;

Dynamic:  $a = 3$  ;

Static scope 中，所以定義的 a 變數只在 main 有效(globe)；dynamic scope 中，則會在堆疊中往上一層找區域變數(local)。