

《程式語言》

一、請就解釋方式（組譯、直譯、編譯）、程式結構（程序導向、物件導向）、標記文字（是、否）等特性，分類說明程式語言C, CSS, C#, HTML, Java, PHP, Python, SQL的特性。請繪製表格作答。（25分）

試題評析	本題為程式語言分類之題目，相似於102年高考、103年檢事官考題，但較為結構化，答案偏向制式。依照題意須將程式語言根據實作方式、程式結構及是否為標記語言區分，唯出現如CSS、Python等較少在考試中出現的語言，對沒有實際經驗的同學而言可能較為吃力，但程度中等的同學應可拿到一半以上的分數。
考點命中	《程式語言》第六回，金乃傑編撰，頁63-67。 《程式語言考前15題猜題》，金乃傑編撰，第15題。

答：

依照題意繪製程式語言特性比較表格如下：

	解釋方式	程式結構	標記文字
C	編譯	程序導向	否
CSS	直譯	非程序導向（宣告式）	否（但嵌入在HTML時必須在<style>標籤中）
C#	編譯	物件導向	否
HTML	直譯	非程序導向（DOM結構）	是
Java	編譯（半直譯、混合式編譯）	物件導向	否
PHP	直譯	物件導向	是（程式碼必須在<?php ?>標籤中）
Python	直譯	物件導向	否
SQL	直譯	非程序導向（4GL）	否

二、請使用如下的BNF語法規則（S為起始符號）：（20分）

S→NP VP|VP

NP→ADJ NP|N

VP→ADV VP|V NP|V

N→考|校|評鑑|成績

V→通過|獲得|成功|失敗

Adj→高|普|特|好|壞

Adv→一定|可能

繪製(1)特校可能通過評鑑

(2)普考一定獲得好高成績

相關推導過程的Parse Tree。

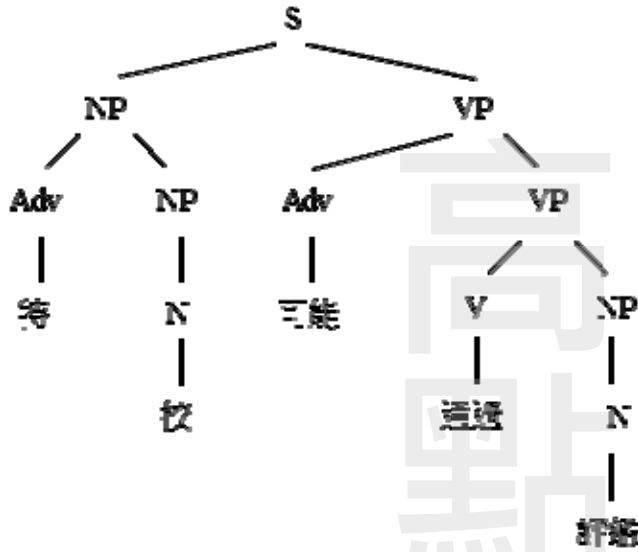
試題評析	本題繪製BNF剖析樹，相似於101鐵路、101地特、104身心障礙考題，此題難度不高，但由於使用「中文」文法，題目較少見且複雜度較運算式，仍必須細心繪製，以免馬前失蹄。程度中等以上的學生應能拿到滿分。
考點命中	《程式語言》第六回，金乃傑編撰，頁5-6。 《程式語言考前15題猜題》，金乃傑編撰，第14題。

【版權所有，重製必究！】

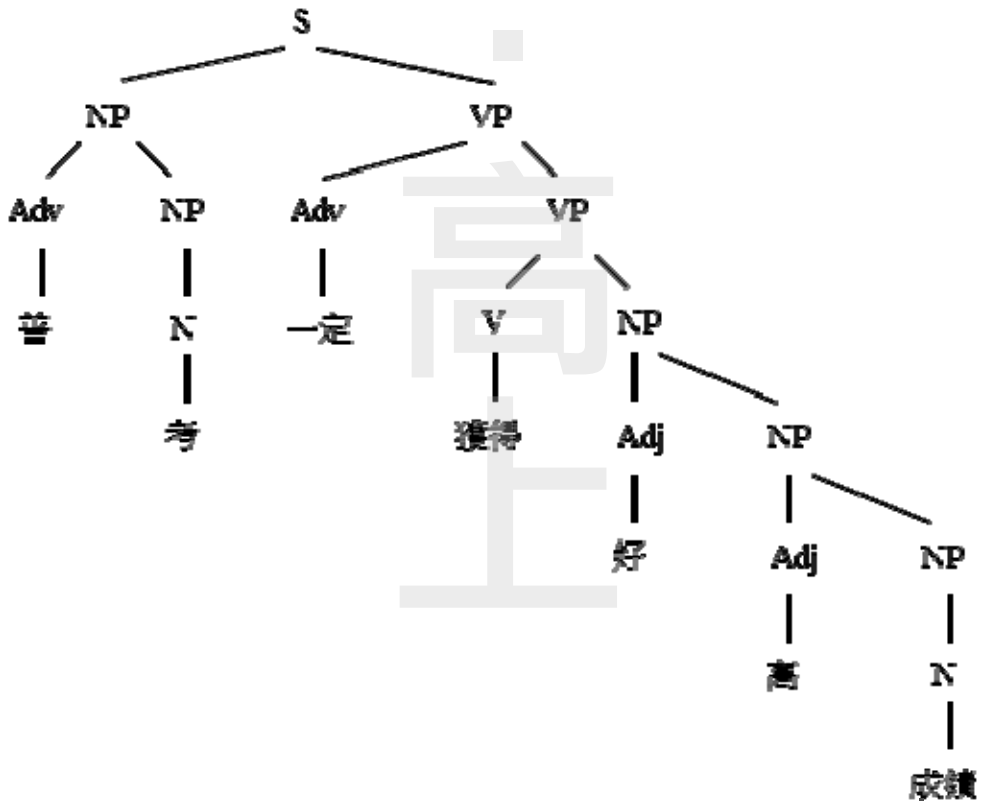
答：

依照題意繪製2顆剖析樹如下：

(一) 語句「特校可能通過評鑑」：



(二) 語句「普考一定獲得好高成績」：



【版權所有，重製必究！】

三、假設現有如下兩張資料表用來存放考生個人訊息及其成績，考生個人訊息，包括代號 (TesterID)、姓名 (Name)，成績 (Grade) 則有五種科目 (1 中文、2 英文、3 數學、4 自然、5 社會) 及成績的等級有三種 (ABC 三種)。請使用 SQL 回應如下的問題。(每小題 5 分，共 20 分)

Tester	Grade
TesterID	TesterID (FK)
Name	SubjectID
	GradeData
	Subject

(一) 請列出中文成績為 A 所有考生的代號及姓名。

(二) 請列出各科不同成績等級的個數，個數使用別名 numCount 來表示，如中文 A5, 中文 B2, ...。

(三) 加入資料 TesterID 6 名為 Oscar Lin 的考生資料及其成績 (中文 A, 英文 A, 數學 C, 自然 B, 社會 A)

(四) 修改 TesterID 5 考生的數學成績為 A。

試題評析	本題為 SQL 語法題目，若同學同時準備資料庫，應能得心應手。
考點命中	《程式語言》第六回，金乃傑編撰，頁 55，上課補充。

答：

(一)

```
SELECT TesterID, Name
FROM Tester t, Grade g
WHERE
g.Subject = '中文' AND g.GradeData = 'A'
AND t.TesterID = g.TesterID
```

(二)

```
SELECT Subject, GradeData, COUNT(*) AS numCount
FROM Grade
GROUP BY Subject, GradeData
ORDER BY Subject ASC, GradeData DESC
```

(三)

```
INSERT INTO Tester(TesterID, Name)
VALUES (6, 'Oscar Lin');
```

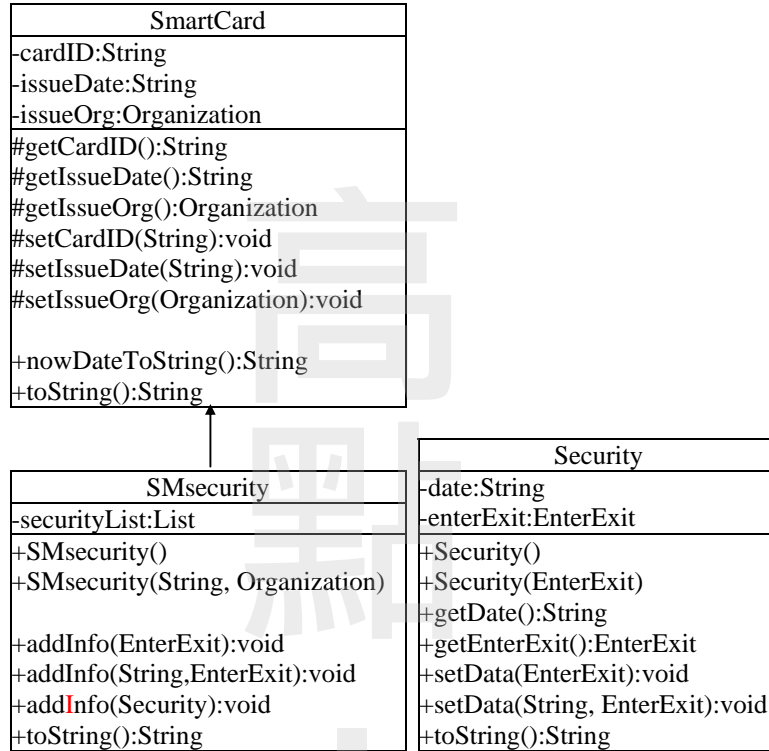
```
INSERT INTO Grade (TesterID, SubjectID, GradeData, Subject)
VALUES (6, 1, 'A', '中文'), (6, 1, 'A', '英文'), (6, 1, 'C', '數學'), (6, 1, 'B', '自然'), (6, 1, 'A', '社會');
```

(四)

```
UPDATE Grade
SET GradeData = 'A'
WHERE
TesterID = 5 AND Subject = '數學'
```

【版權所有，重製必究！】

四、請使用物件導向程式語言（C#或Java），設計一具門禁功能的智慧卡管理資訊系統。系統應依如下的class diagram建立相關的類別資訊。（35分）



- 所有的類別變數皆為private，SMsecurity類別繼承自抽象類別SmartCard
- 發卡單位Organization、進出入狀況EnterExit，請分別使用enum的格式來表示
- 記錄進出入狀況，日期資料若未提供，則填入目前系統的日期時間
- 一張智慧卡，會有多次的進出入記錄，資料值應存放於List線性資料結構中
- 改寫系統的toString()方法後，執行如下的測試動作

```

SMsecurity sms = new SMsecurity("SM001", Organization.NewTaipeiMRT);
sms.addInfo("2015/06/15 18:36:01", EnterExit.Enter);
sms.addInfo("2015/06/15 20:16:01", EnterExit.Exit);
sms.addInfo("2015/06/20 18:20:01", EnterExit.Enter);
sms.addInfo(EnterExit.Exit);
/* for Java */System.out.println(sms.toString());
/* for C# */Console.WriteLine(sms.ToString());
應能產生類似如下的結果
    
```

```

*****Smart card SM001 (6/21/2015 10:38:38 PM, NewTaipeiMRT)
----Enter and Exit information:
(2015/06/15 18:36:01, Enter)
(2015/06/15 20:16:01, Exit)
(2015/06/20 18:20:01, Enter)
(6/21/2015 10:38:38 PM, Exit)
    
```

試題評析	本題為物件導向程式設計題目，相似於100、101年地特，必須從類別圖中的線索了解程式結構、保護等級，並搭配題目中的要求完成程式。本程式有兩處需要注意，其一為在父類別使用private屬性，故設定時必須使用父類別中protected方法，無法直接在子類別設定。其二為大量使用
-------------	--

	overload，類別中存在大量同名函數，程式會根據參數選擇執行的函數，並透過overload實作「預設值」。若同學充分準備，除了較少使用的enum、List及Date外，應都能得心應手，估計程度中等的同學仍應能拿到20分以上。
考點命中	《程式語言》第四回，金乃傑編撰，頁69-76。 《程式語言考前15題猜題》，金乃傑編撰，第9題。

答：

以Java撰寫如下：

```

3 import java.util.ArrayList;
4 import java.util.Date;
5 import java.util.List;
6
7 enum Orgnization{TaipeiMRT, NewTaipeiMRT, TaiChungBRT, KaoHsiungMRT}
8 enum EnterExit{Enter, Exit}
9
10 abstract class SmartCard{ //抽象類別
11     private String cardID; //private屬性
12     private String issueDate;
13     private Orgnization issueOrg;
14
15     protected String getCardID(){return cardID;};
16     protected String getIssueDate(){return issueDate;};
17     protected Orgnization getIssueOrg(){return issueOrg;};
18     protected void setCardID(String cardID){this.cardID = cardID;};
19     protected void setIssueDate(String issueDate){this.issueDate = issueDate;};
20     protected void setIssueOrg(Orgnization issueOrg){this.issueOrg = issueOrg;};
21
22     public abstract String nowDateToString();
23     public abstract String toString();
24 }
25
26 class SMsecurity extends SmartCard{
27     private List<Security> securityList; //儲存紀錄之串列
28
29     SMsecurity(){this("testCard", Orgnization.TaipeiMRT);}
30     SMsecurity(String cardID, Orgnization issueOrg){
31         this.setCardID(cardID); //使用父類別方法設定屬性
32         this.setIssueOrg(issueOrg);
33         securityList = new ArrayList<Security>(); //初始化串列
34     }
35

```

【版權所有，重製必究！】

```

26 class SMsecurity extends SmartCard{
27     private List<Security> securityList;           //儲存紀錄之串列
28
29     SMsecurity(){this("testCard", Organization.TaipeiMRT);}
30     SMsecurity(String cardID, Organization issueOrg){
31         this.setCardID(cardID);                 //使用父類別方法設定屬性
32         this.setIssueOrg(issueOrg);
33         securityList = new ArrayList<Security>(); //初始化串列
34     }
35
36     public String nowDateToString() {             //將日期轉成字串
37         Date dt = new Date();
38         return dt.toString();
39     }
40
41     public String toString() {                   //override輸出字串
42         String returnStr = "*****Smart card " + getCardID() + "(" + nowDateToString() +
43             ", " + getIssueOrg() + ")\nEnter and Exit information:\n";
44         for(Security sec: securityList){        //依序輸出串列內容
45             returnStr += sec.toString()+"\n";
46         }
47
48         return returnStr;
49     }
50
51     public void addInfo(EnterExit enterExit){
52         Security sec = new Security();
53         sec.setData(enterExit);
54         securityList.add(sec);
55     }
56
57     public void addInfo(String date, EnterExit enterExit){
58         Security sec = new Security();
59         sec.setData(date, enterExit);
60         securityList.add(sec);
61     }
62
63     public void addInfo(Security sec){ securityList.add(sec);}
64 }
65

```

【版權所有，重製必究！】

```

66 class Security{
67     private String date;
68     private EnterExit enterExit;
69
70     Security(){
71         Date dt = new Date();
72         date = dt.toString(); //以目前時間為預設值
73     }
74
75     Security(EnterExit enterExit){
76         this();
77         this.enterExit = enterExit;
78     }
79
80     public String getDate(){ return date;}
81     public EnterExit getEnterExit(){ return enterExit;}
82     public void setData(EnterExit enterExit){ this.enterExit = enterExit;}
83     public void setData(String date, EnterExit enterExit){
84         this.date = date;
85         this.enterExit = enterExit;
86     }
87
88     public String toString(){
89         return "("+ date +", " + enterExit + ")";
90     }
91 }
92
93 public class K_104_4 {
94     public static void main(String[] args){
95         SMsecurity sms = new SMsecurity("SM001", Orgnization.NewTaipeiMRT);
96         sms.addInfo("2015/06/15 18:36:01", EnterExit.Enter);
97         sms.addInfo("2015/06/15 20:16:01", EnterExit.Exit);
98         sms.addInfo("2015/06/20 18:20:01", EnterExit.Enter);
99         sms.addInfo(EnterExit.Exit);
100
101         System.out.println(sms.toString()); //測試輸出
102     }
103 }

```

執行結果如下：

```

*****Smart card SM001(Wed Jul 15 08:41:46 CST 2015, NewTaipeiMRT)
Enter and Exit information:
(2015/06/15 18:36:01, Enter)
(2015/06/15 20:16:01, Exit)
(2015/06/20 18:20:01, Enter)
(Wed Jul 15 08:41:46 CST 2015, Exit)

```

【版權所有，重製必究！】