**Java**

1. 請寫出main方法

Public static void main(String args[]){

};

1. 請以Java寫出九九乘法

**public** **class** NineCrossNine {

**public** **static** **void** main(String[] args) {

**for**(**int** i=1;i<=9;i++) {

**for** (**int** j = 1; j <=9; j++) {

System.***out***.printf("%dX%d=%2d ",j,i,i\*j);

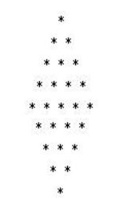
}System.***out***.println();

}

}

}

1. 請用程式畫出



**public** **class** Draw {

**public** **static** **void** main(String[] args) {

**int** end = 10;

**int** half = end/2;

**for** (**int** i = 1; i <= end; i++) {

**if** (i <= half) {

**for** (**int** k = 5; k > i; k--) {

System.***out***.print(" ");

}

**for** (**int** j = 0; j < i ; j++) {

System.***out***.print("\* ");

}

}

**else** {

**for** (**int** k = half; k < i; k++) {

System.***out***.print(" ");

}

**for** (**int** j = 0; j < (end-i); j++) {

System.***out***.print("\* ");

}

}

System.***out***.println("");

}

}

}

1. 請寫出List 、set 、Map三大特性

List 中的元素是有序排列的，每個元素都有一個索引，可以透過索引來存取和修改元素，元素可以被增加、刪除或修改，也允許重複值。

Set 中的元素沒有特定的順序，不像 List 那樣有索引，其中的元素不可修改，也不允許重複，但可以刪除與新增。

Map 是由一組鍵和相應的值組成的集合，每個鍵都唯一且與一個值相關聯。

其中的鍵和值可以被增加、刪除或修改。

1. 物件的三大特性，並說明三個特性是什麼
2. 封裝

將類別屬性設為私有(private)，使得外部(類別外)僅能透過設計者提供的方法來存取，通常為getter()、setter() 以及toString()。

1. 繼承

當一個類別的屬性與方法有包含另一類別所定義的屬性與方法，就可以通過繼承來調用，Extends 子類別會繼承父類別的屬性與方法，例如狗繼承動物，都有生命，都會移動，但每種動物的移動方式不同，狗這個子類別可能就需要方法複寫。

1. 多形

多型與繼承是相輔相成的關係，多行意味著一個物件可以用多種形態來看待，多型的實作仰賴物件之間的繼承關係。

例如:將狗看待成動物。

**Sql**





1. 請用sql語法查出store\_information table 語法

SELECT \*FROM store\_information table;

1. 請新增store\_information語法

INSERT INTO store\_information (store\_Name, Sales,Txn\_Date) VALUES

(‘New York’,2000,’09-Jan-1999’)

1. Geography做group by群組

SELECT Region\_Name,count(\*) as storeAmount

FROM geography

Group By Region\_Name;

1. Geography、store\_information二張table做join

SELECT \*

FROM store\_information

left join geography on store\_information.store\_Name=geography.store\_Name;

SELECT \*

FROM store\_information

right join geography on store\_information.store\_Name=geography.store\_Name;

SELECT \*

FROM store\_information

inner join geography on store\_information.store\_Name=geography.store\_Name;

1. Left join 、right join 、inner join差別

以此題為例

Left join 會以左邊的資料表(store\_information)的(store\_Name)為基去查詢在右邊資料表(geography)的(store\_Name)值,將在左表有的(store\_Name)資料合併到查詢表中

right join 會以右邊的資料表(geography)的(store\_Name)為基去查詢在左邊資料表(geography)的(store\_Name)值,將在右表有的(store\_Name)資料資料合併到查詢表中，若上面沒執行新增紐約的商店，此時的紐約商店資訊就會為Null。

inner join 會將雙邊的資料表都有的(store\_Name)為基來交互查詢並合併到查詢表中。