

考試別：一般警察人員考試

等別：三等考試

類科別：警察資訊管理人員

科目：物件導向程式設計

考試時間：2小時

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

一、表一為用來實現兩數運算的 C# 程式碼，Operation.cs 檔案定義了三個類別；IFactory.cs 定義了一個介面及兩個類別；Program.cs 則定義了 Program 類別。

表一：

Operation.cs	IFactory.cs
<pre> namespace FactoryPattern { class Operation { private double _num1; private double _num2; public double Num1 { get { return _num1; } set { _num1 = value; } } public double Num2 { get { return _num2; } set { _num2 = value; } } public virtual double Result() { return 0; } } class OperationAdd : Operation { public override double Result() { return Num1 + Num2; } } class OperationMultiply : Operation { public override double Result() { return Num1 * Num2; } } } </pre>	<pre> namespace FactoryPattern { interface IFactory { Operation Create(); } class AddFactory : IFactory { public Operation Create() { return new OperationAdd(); } } class MultiplyFactory : IFactory { public Operation Create() { return new OperationMultiply(); } } } </pre>
	Program.cs
	<pre> namespace FactoryPattern { class Program { static void Main(string[] args) { IFactory operFactory = new AddFactory(); Operation oper = operFactory.Create(); oper.Num1 = 3; oper.Num2 = 5; Console.WriteLine(oper.Result()); } } } </pre>

請根據表一回答以下問題：

(一)在 C# 語言中，關鍵字 namespace 的主要用途為何？(2分)

(二)程式進入點在那個類別的那個方法中？(2分)

(三)關鍵字 virtual 與 override 的主要用途為何？(2分)

(四)class AddFactory : IFactory{...} 的意義為何？(2分)

(五)IFactory operFactory = new AddFactory(); 這段程式碼用到了物件導向程式設計三大特性中的那一個特性？(2分)

(六)請依據表一的程式碼畫出 UML 類別圖。(15分)

(請接第二頁)

考試別：一般警察人員考試
等別：三等考試
類科別：警察資訊管理人員
科目：物件導向程式設計

二、佇列（Queue）是一種先進先出（First in first out, FIFO）的資料結構，我們想使用 C++ 語言及陣列來實作一個 QueueAsArray 類別，此類別宣告於 QueueAsArray.h 檔案中，程式碼如表二所示。表三為主函式 main() 的程式碼。

表二：QueueAsArray 類別宣告的程式碼

```
QueueAsArray.h
class QueueAsArray
{
public:
    QueueAsArray(int size);
    ~QueueAsArray();
    void enqueue(double data); //增加一個元素到 Queue的後方
    void dequeue();           //移除Queue的第一個元素
    void display();           //列印Queue中的所有元素
private:
    int front;                 //Queue第一個元素在陣列的位置
    int rear;                  //Queue最後一個元素在陣列的位置
    int size;                  //Queue中一共有多少個位置
    double* array;            //用來儲存Queue元素的陣列之指標
    bool isFull();            //檢查Queue是否已經被填滿
    bool isEmpty();           //檢查Queue是否是空的
};
```

表三：主函式 main() 的程式碼

```
int main() {
    QueueAsArray* queue = new QueueAsArray(3);
    queue->enqueue(3);
    queue->enqueue(1);
    queue->enqueue(2);
    queue->enqueue(5);
    queue->enqueue(9);
    queue->dequeue();
    queue->enqueue(4);
    queue->display();
    system("pause");
    return 0;
}
```

enqueue() 的功能是把元素加入 Queue 的後方，dequeue() 的功能是移除 Queue 的第一個元素，display() 是列印 Queue 中所有的元素，圖一描述 QueueAsArray 中 enqueue 與 dequeue 的動作原理以及變數 front 及 rear 之改變情況。

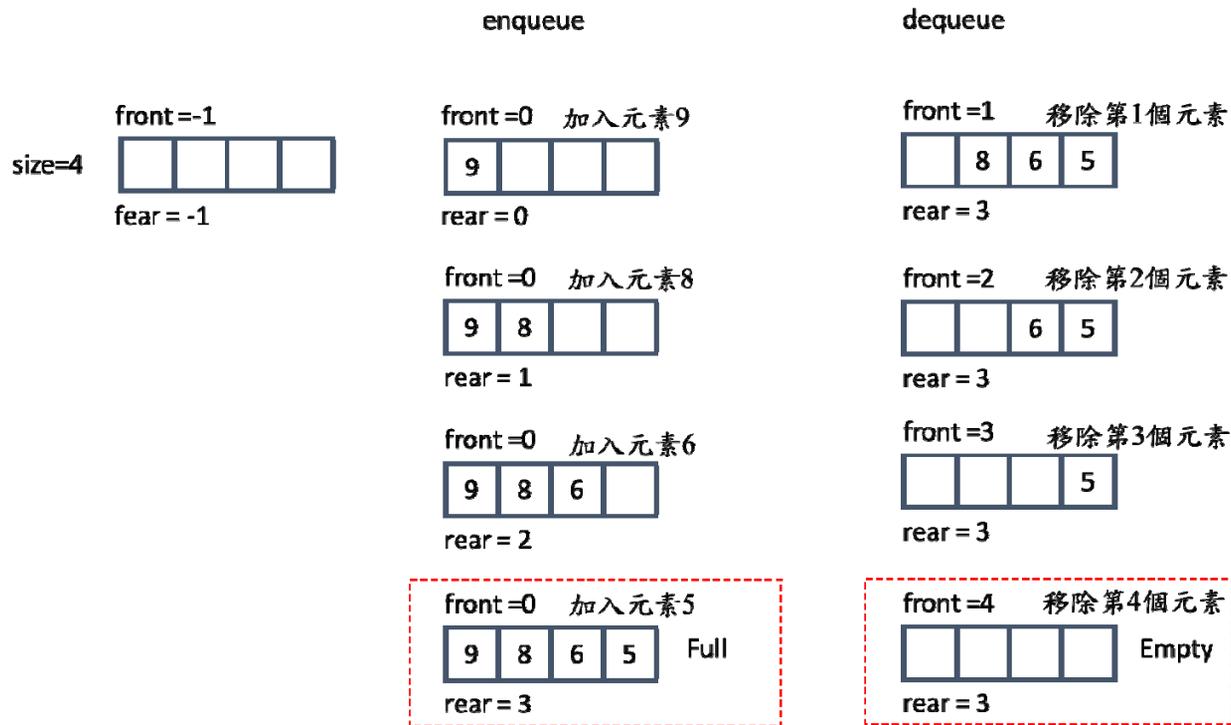
(一) 請以 C++ 程式語言實現 QueueAsArray(int size)、~QueueAsArray()、enqueue()、dequeue()、display()、isFull()、isEmpty() 之功能，並將此程式碼存於 QueueAsArray.cpp 中，若物件使用到動態記憶體配置，則在執行解構子時要釋放該記憶體空間。(20 分)

(二) 根據表三，請問指標 queue 指向的佇列物件可儲存幾個元素？(2 分)

(三) 根據表三，請問程式執行完畢後，佇列內容為何？(3 分)

(請接第三頁)

考試別：一般警察人員考試
等別：三等考試
類科別：警察資訊管理人員
科目：物件導向程式設計



圖一：在 Queue 中增加元素或刪除元素時，front 及 rear 之變化

考試別：一般警察人員考試
等別：三等考試
類科別：警察資訊管理人員
科目：物件導向程式設計

三、表四的程式碼是以多型的概念來實作圓形以及正方形的面積計算，請依據以下的要求進程式碼的改寫：

- (一)將 Shape 類別修改成抽象類別。(3 分)
- (二)用常數函式的概念來避免在執行 getArea() 函式時，修改到 Circle 或 Square 物件成員的值。(3 分)
- (三)在 Circle 類別中，有一個圓周率 PI 的資料成員，這份程式碼的缺陷是，每生成一個 Circle 物件，就需要一塊記憶體來記錄 PI 值，造成記憶體的浪費，請用 static 成員的概念來改善 Circle 類別的程式碼。(3 分)
- (四)當幾何參數小於等於 0 時，執行 Circle 或 Square 的建構子時，物件依然能夠生成，但面積的計算會出錯，例如，當傳入的參數是 -2 時，生成的 Square 物件，執行 getArea() 後會傳回 4。這是類別設計上的瑕疵，為了改善這個狀況，應該利用靜態 (static) 方法的概念設計一個 Create() 的函數，當幾何參數小於等於 0 時，傳回 Null 指標，而當幾何參數大於 0 時，則呼叫該類別的建構子。(6 分)

表四: Shape, Circle, Square 類別的程式碼

Shape.h	Shape.cpp
<pre>class Shape { public: virtual double getArea() { return 0; }; }; class Circle : Shape { private: double const PI = 3.1415926; double _radius; public: Circle(double radius); double getArea(); }; class Square : Shape { private: double _length; public: Square(double length); double getArea(); };</pre>	<pre>#include "Shape.h" #include <stdio.h> Circle::Circle(double r) { _radius = r; } double Circle::getArea() { return PI*_radius*_radius; } Square::Square(double length) { _length = length; } double Square::getArea() { return _length*_length; }</pre>

- (五)在表五中我們產生 10 個亂數，亂數的範圍是從 -3 到 6，我們把這 10 個亂數儲存在 data 陣列中，接下來我們利用 data 陣列的資料來產生 Circle 及 Square 物件，當陣列的 index 是偶數時產生 Circle 物件，奇數時產生 Square 物件，將這些物件以父類別指標 Shape* 的形式記錄下來，並將其加入到 vector 容器中。用 vector 的 iterator 功能將 vector 所有的物件取出計算其面積並加總，而後輸出結果。請以 C++ 語言完成表五的程式碼。(15 分)

(請接第五頁)

考試別：一般警察人員考試
等別：三等考試
類科別：警察資訊管理人員
科目：物件導向程式設計

表五：第三題的主函式

```
int main()
{
    Shape* s;
    宣告 vector 類別的變數 v，可用來儲存 Shape*。(2分)
    int data[10];
    for (int i = 0; i <= 9; i++)
    {
        data[i] = rand() % 10 - 3; // 亂數範圍為 -3 到 6
    }
    for (int i = 0; i <= 9; i++)
    {
        if(i%2==0)
            利用 Circle 的靜態方法 Create，生成 Circle 物件，並將 Circle* 轉
            型為 Shape*。(2分)
        else
            利用 Square 的靜態方法 Create，生成 Square 物件，並將 Square*
            轉型為 Shape*。(2分)
    }
    將上面產生的 Shape* 加入 vector 容器 v 中。(3分)
    }
    double sum = 0; //計算面積加總的變數
    利用 vector 的 iterator, begin(), end() 等功能將容器 v 的 Shape*
    拿出來進行計算面積，並求其加總。特別要注意的是，必須檢查
    Shape* 是否為 NULL。(6分)
    cout << sum << endl;
    system("pause");
}
```

四、複數 z 的極座標表示法為 (r, θ) ，其中 r 為複數 z 到原點的距離，將 z 與原點連成的一直線，此直線與實數軸的夾角為 θ 。我們可以將複數 z 表成 $z = (r, \theta)$

$$\text{若 } z_1 = (r_1, \theta_1), z_2 = (r_2, \theta_2), \text{ 則複數的除法可以表成 } \frac{z_2}{z_1} = \left(\frac{r_2}{r_1}, \theta_2 - \theta_1 \right)$$

請利用 C++ 程式語言實現複數極座標表示法的除法運算，程式之撰寫請依據以下要求：

- (一) 複數之類別名稱為 Complex，有兩個 private 等級的資料成員，分別是 `_radius` 以及 `_angle`，`_radius` 代表 r ，`_angle` 代表 θ 。(3分)
- (二) 請用 inline 函數來實現資料成員的存取。(3分)
- (三) 實作運算子 `/=` (註 $z_2 /= z_1$ 等同於 $z_2 = z_2 / z_1$)。(4分)
- (四) 請設計一個 ComplexException 類別用來進行除零的例外處理，ComplexException 必須繼承 `std::exception` 並複寫 `what()` 函式，當除 0 的狀況發生時，丟出 DividedByZero 字串。(5分)
- (五) 請撰寫 main 函式來驗證程式之功能。(5分)