

《程式設計概要》

一、請回答以下對C/C++程式語言的敘述正確與否？如果敘述是錯誤的，需說明正確敘述為何。（每小題3分，共30分）

- (一)程式執行的結果與預期的不同，這種錯誤稱為執行期錯誤（Runtime Error）。
- (二)屬自由格式（Free-Form）、隱含型別宣告（Implicit Type Declaration）程式語言。
- (三)switch選擇結構中，default敘述為選擇性，故可省略。
- (四)陣列（Array）無法存放一連串資料型別不同的相關資料。
- (五)當&&運算子中的只要有一個運算元為真時，則該運算式的結果為真。
- (六)可使用結構（Structure）陣列變數的名稱指向該陣列中的任一個資料。
- (七)程式在編譯階段即配置完成其所需記憶體的方式稱為動態記憶體配置（Dynamic Memory Allocation）。
- (八)當不同的物件收到相同的訊息時，會以各自的方式作處理的特點稱為多載（Overloading）。
- (九)子類別將繼承自父類別的方式重新定義，而且不會影響到父類別的方法，這個特點稱為多載（Overloading）。
- (十)子類別可多重繼承自父類別內所有的資料成員與成員函數。

試題評析	本題測試C/C++的語法與C++的物件導向觀念，尤其是多載(Overloading)與覆寫(Overriding)，英文單字非常相近，容易混淆，觀念清楚可拿高分。
考點命中	《高點·高上程式設計概要講義》，許振明編撰，第一回頁13~17、頁27、頁36~40、頁93~97、頁149~154；第二回頁77~78、頁105~106、頁110~119、頁152~153。

答：

- (一)False, 執行結果與預期不同是邏輯上的錯誤(Login Error)。
- (二)False, C/C++非隱含型別宣告，必須使用前先宣告變數的型態。
- (三)True, switch 的default不是必要關鍵字，所以可省略。
- (四)True, 陣列只能儲存相同型態的變數。
- (五)False, &&(AND)有一邊為真時，由另一邊決定最後答案。
- (六)False, 需使用指標變數。
- (七)False, 動態記憶體配置是在執行時完成配置記憶體空間。
- (八)False, 多載(Overloading)指的是相同函數名稱可傳入不同的訊息個數或形態。
- (九)False, 重新定義函數稱覆寫(Overriding)。
- (十)True, C++支援多重繼承。

【版權所有，重製必究！】

二、請問以下程式輸出的結果為何？（10分）

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(void){
    int i=0, sum=0;
    while(i < 21) {
        if(i % 3 != 0) {
            i++;
            continue;
        }
        sum += i++;
    }
    cout<< "i=" <<i<< ", sum=" << sum <<endl;

    return 0;
}
```

試題評析	本題測試迴圈的規則與continue的用法，注意上課老師的重點提示就可取分。
考點命中	《高點·高上程式設計概要講義》第一回，許振明編撰，頁43～92。

答：

```
int main(void){
    int i=0, sum=0;
    while(i<21){
        if(i%3 !=0){
            i++;
            continue;
        }
        sum+=i++; // 3的倍數才執行加法，所以本題答案 3+6+9+12+15+18=63
    }
    cout << "i=" << i << ", SUM=" << sum << endl;

    return 0;
}
```

i=21, SUM=63

【版權所有，重製必究！】

三、請問以下程式輸出的結果為何？（10分）

```
#include <iostream>
#include <cstdlib>
using namespace std;
void fun1(int, int);
void fun2(int *, int *);
void fun3(int *, int, int);
int main(void){
    int a[]={1, 2, 3, 4, 5}, i;
    fun1(a[1], a[3]);
    fun2(a, a+2);
    fun3(a, 5, 0);
    cout<< "執行所有函數後，陣列的內容為:" << endl;
    for(i=0, i<5; i++)
        cout<< "a[" << i << "]= " << a[i] << endl;

    return 0;
}

void fun1(int x, int y){
    int tmp;
    tmp=y;
    y=x;
    x=tmp;
}

void fun2(int *x, int *y){
    int tmp;
    tmp=*y;
    *y=*x;
    *x=tmp;
}

void fun3(int *ptr, int n, int num){
    *(ptr+n-1)=num;
}
```

試題評析

本題測試函數的傳遞參數規則，主要測試call by value與call by address。這是C/C++的基本傳遞方法，所以取分不難。

考點命中

《高點·高上程式設計概要講義》第一回，許振明編撰，頁98~102。

答：

```

void fun1(int, int);
void fun2(int *, int *);
void fun3(int *, int, int);
int main(void){
    int a[]={1,2,3,4,5}, i;
    fun1(a[1], a[3]);      // call by value, a[1]與a[3]不交換
    fun2(a, a+2);         // call by address a[0]與a[2]交換
    fun3(a, 5, 0);        // a[4]=0
    cout << "執行所有函數後，陣列的內容為:" << endl;
    for(i=0; i<5; i++)
        cout << "a[" << i << "]=" << a[i] << endl;

    return 0;
}
void fun1(int x, int y){
    int tmp;
    tmp=y;
    y=x;
    x=tmp;
}
void fun2(int *x, int *y){
    int tmp;
    tmp=*y;
    *y=*x;
    *x=tmp;
}
void fun3(int *ptr, int n, int num){
    *(ptr+n-1)=num;
}

```

執行所有函數後，陣列的內容為：

```

a[0]=3
a[1]=2
a[2]=1
a[3]=4
a[4]=0

```

【版權所有，重製必究！】

四、以下是一個完整的Python程式，請問執行後的結果為何？（10分）

```
for i in range(1, 7): #產生一個1, 2, 3, 4, 5, 6的整數序列
    for j in range(i, 1, -1): #-1表示階值(step value)
        print(" ", end=" ")

    for j in range(i, 7-i+1):
        print(j, end=" ")

    print()
```

試題評析

這是今年度第一次出的python問題，搭配AI與machine learning的主題，是非常夯的主題。本題測試迴圈用法，只要熟悉C/C++的迴圈語法，要取分是容易的。

考點命中

《高點·高上程式設計概要講義》第一回，許振明編撰，頁43~92。

答：

```
for i in range(1,7):
    for j in range(i,1,-1):
        print(" ", end=" ")

    for j in range(1, 7-i+1):
        print(j, end=" ")

    print()

1 2 3 4 5 6
  1 2 3 4 5
    1 2 3 4
      1 2 3
        1 2
          1
```

五、撰寫一個完整的C/C++、Java或Python程式，讓使用者輸入姓名（name）、小考成績（quiz）、期中考成績（midterm）與期末考成績（final）後，將這三個成績傳入呼叫函數calScore()，該函數需傳回所計算後的學期成績；程式再將學期成績傳入呼叫函數calGrade()，該函數需傳回成績等第結果；最後程式需從螢幕顯示出使用者所輸入的姓名、學期成績與成績等第結果。程式顯示結果後，需再詢問使用者是否繼續輸入，當使用者輸入'Y'或'y'時則繼續輸入，否則結束離開程式。（20分）

學期成績計算公式為：小考成績*0.15+期中考成績*0.35+期末考成績*0.5

成績等第標準如下：

學期成績	成績等第
學期成績 ≥ 90	A
89 ≥ 學期成績 ≥ 80	B
79 ≥ 學期成績 ≥ 70	C
69 ≥ 學期成績 ≥ 60	D
學期成績 ≤ 59	F

【版權所有，重製必究！】

試題評析	本題是C語言的綜合試題，主要以題目描述的內容撰寫程式。主要規則包含條件、迴圈與函數呼叫，平常用功的同學取分不難。
考點命中	《高點·高上程式設計概要講義》第一回，許振明編撰，頁30~92；頁98~102。

答：

```

#include <iostream>
#include <cstdio>

using namespace std;

double calScore(int quiz, int midterm, int final){
    return quiz*0.15+midterm*0.35+final*0.5;
}

char *calGrade(double score){
    if(score>=90)
        return "A";
    else if(score>=80)
        return "B";
    else if(score>=70)
        return "C";
    else if(score>=60)
        return "D";
    else
        return "F";
}

int main(void){
    int quiz, midterm, final;
    double score;
    char *result, name[100];
    char ch='n';

    do{
        cout << "enter name quiz midterm final:" << endl;
        cin >> name >> quiz >> midterm >> final;

        score = calScore(quiz, midterm, final);
        result = calGrade(score);

        cout << "name: " << name << ", score: " << score << ", result: " << result << endl;

        cout << "play again?(y/n)" << endl;
        cin >> ch;
    }while(ch=='Y' || ch=='y');

    return 0;
}

```

【版權所有，重製必究！】

六、撰寫一個完整的C/C++、Java或Python程式，以物件導向方式設計一個名為transcript的類別，此類別包括以下屬性與方法：（20分）

- name代表姓名的私有（private）字串資料項目
- quiz代表小考成績的私有數值資料項目
- midterm代表期中考成績的私有數值資料項目
- final代表期末考成績的私有數值資料項目
- 建構函數可將新建立物件之姓名作初始化
- 設計calScore()公有方法計算學期成績（score）的私有數值資料項目
- 設計calGrade()公有方法計算成績等第（grade）的私有字元資料項目
- 設計printReport()方法從螢幕顯示出學生姓名、學期成績與成績等第結果
- 學期成績計算公式與成績等第標準同第五題。

程式需設計迴圈讓使用者輸入資料以驗證程式的正確性，直到使用者輸入'n'或'N'時結束程式。

試題評析	本題與以前一樣，主要測試物件導向程式觀念，透過题目的引導，利用class封裝資料與函數成員，最後使用主程式執行呼叫流程。
考點命中	《高點·高上程式設計概要講義》第二回，許振明編撰，頁71~120。

答：

```
#include <iostream>
#include <cstring>

using namespace std;

class transcript{
private:
    char *name;
    int quiz;
    int midterm;
    int final;

public:
    transcript(char *name){
        int len = strlen(name)+1;
        this->name = new char[len];
        strcpy(this->name, name);
    }

    double calScore(){
        return quiz*0.15+midterm*0.35+final*0.5;
    }

    char *calGrade(){
        int score=calScore();

        if(score>=90)
            return "A";
        else if(score>=80)
            return "B";
        else if(score>=70)
            return "C";
        else if(score>=60)
```

【版權所有，重製必究！】

```
        return "D";
    else
        return "F";
}

void printReport(){
    cout << "name: " << name << ", score: " << calScore() << ", result: " << calGrade() << endl;
}

void setData(int quiz, int midterm, int final){
    this->quiz=quiz;
    this->midterm=midterm;
    this->final=final;
}
};

int main(void){
    transcript *myObj;
    int quiz, midterm, final;
    double score;
    char *result, name[100];
    char ch='n';

    do{
        cout << "enter name quiz midterm final:" << endl;
        cin >> name >> quiz >> midterm >> final;

        myObj = new transcript(name);
        myObj->setData(quiz, midterm, final);
        myObj->printReport();

        cout << "play again?(y/n)" << endl;
        cin >> ch;
    }while(ch=='Y' || ch=='y');

    return 0;
}
```

【版權所有，重製必究！】