

# 程式設計與實習(二)

## 期末考試卷(A)

2022062413101440

班級: \_\_\_\_\_

姓名: \_\_\_\_\_

學號: \_\_\_\_\_

第一題(16%)	
第二題(6%)	
第三題(8%)	
第四題(30%)	
CPE 一星題(40%)	
課程加分	
總分(最高 100 分)	

【第一題】請寫出下列程式碼輸出結果，如有程式錯誤請將錯誤地方圈起來並於答案區寫出錯誤原因。(16%)

(A)

```
class A{
    int a = 11;
    A(){
    A(int a,int b)
    {
        System.out.print("111");
    }
    void A(int ...a){
        System.out.print(this.a);
        A(1,2);
    }
}
class B extends A{
    int a =22;
    void A(int a, int b)
    {
        System.out.println(this.a);
    }
}
public class q1_A {
    public static void main(String[] args) {
        A b = new B();
        b.A(1,2);
    }
}
```

作答區：

(B)

```
class a_b{
    private int a = 15;
    private int b = 10;
    a_b(int a, int b)
    {
        a = this.a+a;
        b = this.b;
        System.out.print(a+b);
    }
    void a_b(int a, int b){
        a = this.a*2;
        System.out.print(a+this.b);
    }
}
class b_b extends a_b{
    private int a;
    private int b;
    b_b(int a, int b){
        super(a,b);
        this.b = b;
        this.a = this.a+b;
        System.out.print("222");
    }
    void a_b(int a, int b){
        a = this.a*3+b;
        System.out.print(a+this.b);
    }
}
public class q1_B {
    public static void main(String[] args) {
        a_b test = new b_b(10,10);
        test.a_b(20,10);
    }
}
```

作答區：

(C)

```
interface a{
    int a = 5;
    int b = 12;
    void a();}
interface b extends a{
    int a = 15;
    void b();}
class a_c implements b{
    a_c(int a, int c){
        System.out.print(this.a+b);
        b();
    }
    public void a(){System.out.print(a+this.b);
    }
    public void b(){System.out.print(a+b);}
}
class b_c extends a_c {
    private int a = 5;
    private int b = 10;
    b_c(int b, int d)
    {
        super(b,d);
        this.a = b;
        b();
    }
    public void b(){
        System.out.print(this.a*this.b);
    }
}
public class q1_c {
    public static void main(String[] args) {
        b_c test = new b_c(12,10);
        a_c test2 = new b_c(10, 20);
        test.a();
        test2.b();
    }
}
```

作答區：

(D)

```
abstract class a{
    int a=10;
    int b=25;
    void a(){System.out.print(a+b);}
    void a(int a, int b){System.out.print(b-a);}
}
class a_d extends a{
    a_d(int a, int b)
    {
        this.a = a + this.b;
        this.b = this.a + b;
        a();
    }
}
class b_d extends a_d{
    int c = 10;
    b_d(int b, int d)
    {
        super(b, d);
        this.a = b+this.b;
        this.b = d+this.a;
    }
    void a(){
        System.out.print(this.a+c);
    }
}
public class q1_d {
    public static void main(String[] args) {
        a_d test = new a_d(10, 20);
        a_d test2 = new b_d(22,33);
        test.a();
        test2.a();
        test2.a(2,3);
    }
}
```

作答區：

【第二題】問答題 (6%)

(A) 請指出兩點 interface(介面)與 Abstract class(抽象類別)相異之處?(2%)

①	
②	

(B) 選擇題(4%)

```
public class q2_b{
    static void test() throws Error {
        if (true)
            throw new AssertionError();
        System.out.print("test ");
    }

    public static void main(String[] args) {
        try {
            test();
        } catch (Exception ex) {
            System.out.print("exception ");
        }
        System.out.print("end ");
    }
}
```

What is the result?

- A. end
- B. Compilation fails.
- C. exception end
- D. exception test end
- E. A Throwable is thrown by main.
- F. An Exception is thrown by main.

作答區：

Given:

```
11. // insert code here
12. private N min, max;
13. public N getMin() { return min; }
14. public N getMax() { return max; }
15. public void add(N added) {
16. if (min == null || added.doubleValue() <
    min.doubleValue())
17. min = added;
18. if (max == null || added.doubleValue() >
    max.doubleValue())
19. max = added;
20. }
21. }
```

Which two, inserted at line 11, will allow the code to compile?

- A. public class MinMax<N extends Number> {
- B. public class MinMax<? extends Number> {
- C. public class MinMax<N extends Object> {
- D. public class MinMax<?> {
- E. public class MinMax<? extends Object> {
- F. public class MinMax<N extends Integer> {

作答區：

【第三題】請根據題目所示的情境，完成片段程式 (8%)

使用自訂的類別(class)來模擬汽車行駛時所消耗油量的情況，此自訂的類別會紀錄汽車油箱的油量的油量及耗油效率(每公升可行駛的里程數)。並有一個行駛指定里程數的方法，此方法會在油量不足以行駛完指定的里程時，拋出自訂的 NoGasException 例外物件。至於自訂的 NoGasException 例外類別，則只單純繼承 Exception 類別，並未改寫或新增任何成員，輸出結果如右下圖所示。

```
import java.util.*;

① //自行定義 NoGasException 類別
class myCar{
    double GasTank;
    double Efficient;
    public myCar(double g, double e){
        this.GasTank = g;
        this.Efficient = e;
    }
    public void go(double distance) ②
        NoGasException{
            double GasNeed = distance/Efficient;
            GasTank -= GasNeed;
            if(GasTank<0)
                ③
        }
    public double checkGas(){⑦}
}

public class q4{
    public static void main(String[] args) {
        Scanner sca = new Scanner(System.in);
        System.out.println("計算汽車耗油狀況");
        System.out.println("請輸入您的車油量(公升):");
        double g = ④
        System.out.println("請輸入一公升可跑的公里數:");
        double eff = ④
        myCar nissan = new myCar(g, eff);

        do{
            System.out.println("您現在還有多少里程");
            double m = ④
            ⑤{
                nissan.go(m);
                System.out.println("開了"+ m + " 公里後. ");
                System.out.println("還剩 "+ nissan.checkGas() + "
                公升的油");
            }
            ⑥ ( ⑧ ){
                System.out.print("油量不足");
            }
        } while(nissan.checkGas()>=0);
    }
}
```

```
計算汽車耗油狀況
請輸入您的車油量(公升):
10
請輸入您的車一公升可跑的公里數:
5
您現在還有多少里程
30
開了30.0 公里後.
還剩 4.0公升的油
您現在還有多少里程
14
開了14.0 公里後.
還剩 1.2000000000000002公升的油
您現在還有多少里程
10
油量不足%
```

作答區：

①		②	
③		④	
⑤		⑥	
⑦		⑧	

**【第四題】請根據題目所示的情境，完成片段程式 (30%)**

助教小藍每當期末考或時都很懊惱同學們的成績登記和總成績計算，小藍畢竟是一個程設助教，於是他使用物件陣列寫了一個具有統計分數的程式以方便總成績的計算，功能需求以下輸入、輸出說明：

輸入：第一列數字 n 代表以下有多少組測試資料。接著四行為一組測資，分別姓名(name)、程設成績(programing)、微積分成績(Calculus)、英文成績(English)

輸出：成績計算採加權計算(個別原始分數\*加權數的加總/總加權數)，程設加權數為 4、微積分加權數為 2、英文加權為 3，總加權數為 9，輸出結果會根據輸入順序依序印出

範例輸入：

```
3
Mark Lan
50
70
100
Luke Huang
60
80
60
Ray Ho
70
60
55
Alex Huang
76
59
89
```

範例輸出：

```
Mark Lan 程式設計:70 微積分:60 英文:100 總平均:77.0 分
Luke Huang 程式設計:60 微積分:55 英文:80 總平均:65.0 分
Ray Ho 程式設計:70 微積分:60 英文:55 總平均:62.0 分
Alex Huang 程式設計:76 微積分:59 英文:89 總平均:76.0 分
```

```

import java.util.*;

public class q3 {
    static Scanner sc = new Scanner(System.in);
    static String inputScore() {
        String a = ①
        return a;
    }
    static String inputName() {
        String a = ①
        return a;
    }
    public static void main(String[] args) {
        int num = ②
        sc.nextLine();
        ③[] scoreArr = new ③[num];
        for (int i = 0; i < num; i++) {
            scoreArr[i] = ④ Sheet(i + 1,
inputName(), inputScore(), inputScore(), inputScore());
        }
        for (int i = 0; i < num; i++) {
            System.out.println(scoreArr[i]);
        }
        sc.close();
    }
}
⑤ Sheet {
//使用 id 設為公用型態，其他設為私有
    ⑥ int id;
    ⑦ String name;
    ⑦ Programming pScore;
    ⑦ Calculus cScore;
    ⑦ English eScore;
    ⑦ float scoreAvg;
    ⑦ int credit = 9;

    Sheet(int id, String name, String _pScore, String
_cScore, String _eScore) {
        ⑧ .id = id;
        ⑧ .name = name;
        ⑧ .pScore = ④
        Programming(Integer.parseInt(_pScore), 4);

```

```

        ⑧ .cScore = ④
        Calculus(Integer.parseInt(_cScore), 2);
        ⑧ .eScore = ④
        English(Integer.parseInt(_eScore), 3);
        cal();
    }
    @Override
    ⑨ toString() {
        return name + " " + pScore.toString()+
cScore.toString()+ eScore.toString()+"總平均:" +
scoreAvg + "分";
    }
    ⑩ cal() {
//計算總平均，僅能使用 creditMultiplyScore()方法
        scoreAvg = ⑪
    }
    ⑤ Score {
        int score = 0;
        int credit = 1;
        Score(int _score, int _credit) {
            ⑧ .score = _score;
            ⑧ .credit = _credit;
        }
        @Override
        ⑨ toString() {
            return ":" + score;
        }
        public int creditMultiplyScore() {
            return ⑪
        }
    }
    class Programming ⑮ Score {
        Programming(int score, int credit) {
            ⑫
        }
        @Override
        ⑨ toString() {
            ⑭ "程式設計" + ⑬
        }
    }
}

```

```

class Calculus ____⑮____ Score {
    Calculus(int score, int credit) {
        ____⑫____
    }
    @Override
    ____⑨____ toString() {
        ____⑭____ "微積分" + ____⑬____
    }
}

class English ____⑮____ Score {
    English(int score, int credit) {
        ____⑫____
    }
    @Override
    ____⑨____ toString() {
        ____⑭____ "英文" + ____⑬____
    }
}
}

```

作答區：

①	
②	
③	
④	
⑤	
⑥	
⑦	
⑧	
⑨	
⑩	
⑪	
⑫	
⑬	
⑭	
⑮	