

通識計算機程式設計期末考試題參考解答, 6/25/2010

1. 撰寫 C# 敘述達成下列要求: (假設 `using System;` 敘述已經包含於程式中)
- (a) 撰寫類別程式 **Rhombus** 代表菱形, 其中宣告私有 **double** 變數 **d1**、**d2**, 分別代表兩互相垂直的對角線長, 另有建構式設定 **d1**、**d2** 之值, 屬性 **D1**、**D2** 分別傳回與設定 **d1**、**d2** 之值, 此處可先不處理例外情況 (6%)

Ans:

```
class Rhombus
{
    private double d1;
    private double d2;
    public Rhombus(double d1, double d2)
    {
        this.d1 = d1;
        this.d2 = d2;
    }
    public double D1
    {
        set { d1 = value; }
        get { return d1; }
    }
    public double D2
    {
        set { d2 = value; }
        get { return d2; }
    }
}
```

- (b) 撰寫類別程式 **Square** 代表正方形, 繼承 **Rhombus**, 在傳入邊長 **a** 之建構式中, 先呼叫父類別建構式設定 **d1**、**d2** 之值為 $\sqrt{2} a$, 由於設定初值之工作已於父類別建構式完成, 此一 **Square** 建構式主體部份可以是空的(6%)

Ans:

```
class Square : Rhombus
{
```

```

    public Square(double a) :
        base(Math.Sqrt(2) * a, Math.Sqrt(2) * a)
    {
    }
}

```

(c) 撰寫介面 **Shape**，其中定義函式 **Area()**，用以計算面積，傳回 **double** 數值 (3%)

Ans:

```

interface Shape
{
    double Area();
}

```

(d) 改寫(a)的類別程式 **Rhombus**，實作 **Shape**，並完成函式 **Area()** (3%)
[注意菱形面積為 $d1*d2/2$]

Ans:

```

class Rhombus : Shape
{
    private double d1;
    private double d2;
    public Rhombus(double d1, double d2)
    {
        this.d1 = d1;
        this.d2 = d2;
    }
    public double D1
    {
        set { d1 = value; }
        get { return d1; }
    }
    public double D2
    {
        set { d2 = value; }
        get { return d2; }
    }
}

```

```

    public double Area()
    {
        return d1 * d2 / 2;
    }
}

```

(e) 在類別程式 **Rhombus** 的建構式中，檢查輸入參數是否均為正，若有不滿足之狀況發生，**throw** 一個 **ArgumentException** 例外物件 (3%)

Ans:

```

class Rhombus : Shape
{
    private double d1;
    private double d2;
    public Rhombus(double d1, double d2)
    {
        if (d1 <= 0 || d2 <= 0) throw new
            ArgumentException();
        this.d1 = d1;
        this.d2 = d2;
    }
    public double D1
    {
        set { d1 = value; }
        get { return d1; }
    }
    public double D2
    {
        set { d2 = value; }
        get { return d2; }
    }
    public double Area()
    {
        return d1 * d2 / 2;
    }
}

```

(f) 寫一段測試程式，自鍵盤讀入 **d1**、**d2** 之值，再以之建立一個菱形物件

rh 。 用 try-catch 攔截 FormatException 、 ArgumentException、Exception 例外物件，並於例外發生時，要求使用者重新輸入，直到正確為止 (6%)

Ans:

```
class Program
{
    static void Main(string[] args)
    {
        bool successful = false;
        do
        {
            try
            {
                Console.WriteLine("輸入菱形對角線長d1");
                double d1 = int.Parse(Console.ReadLine());
                Console.WriteLine("輸入菱形對角線長d2");
                double d2 = int.Parse(Console.ReadLine());
                Rhombus rh = new Rhombus(d1, d2);
                successful = true;
            }
            catch (FormatException e)
            {
                Console.WriteLine(e);
            }
            catch (ArgumentException e)
            {
                Console.WriteLine(e);
            }
            catch (Exception e)
            {
                Console.WriteLine(e);
            }
        } while (!successful);
    }
}
```

2. 找出以下程式片段之錯誤，並予更正.

(a) (3%) 一個錯誤

```
class B
{
    public static int Triple(int i)
    {
        return 3*i;
    }
}
```

```
class Program
```

```
{
    static void Main(string[] args)
    {
        B b = new B();
        Console.WriteLine(b.Triple(2));
    }
}
```

→靜態函式不可以物件呼叫，須將b改為類別名稱B
亦可此處不動，刪去class B中Triple宣告的 static
關鍵字

(b) (3%) 一個錯誤

```
struct A
```

```
{
    public int i;
    public A() { i = 0; }
    public A(int i) { this.i = i; }
}
```

```
class Program
```

```
{
    static void Main(string[] args)
    {
        A a = new A(2);
        Console.WriteLine(a.i);
    }
}
```

(c) (6%).一項錯誤，請依註解以複製建構式更正

```
class C
{
    private int i;
    public C()
    {
        i = 0;
    }
    public int I
    {
        set { i = value; }
        get { return i; }
    }
    // 此處須加入複製建構式的宣告
}

class Program
{
    static void Main(string[] args)
    {
        C c1 = new C();
        C c2 = new C();

        // 這一行需改為利用複製建構式的寫法
        c2 = c1;

        // 要讓c1和c2的i分別等於10和20
        c1.I = 10;
        c2.I = 20;
    }
}
```

Ans:

```
class C
{
    private int i;
    public C()
```

```

    {
        i = 0;
    }
    public int I
    {
        set { i = value; }
        get { return i; }
    }
    // 此處須加入複製建構式的宣告
    public C(C c)
    {
        this.i = c.i;
    }
}

class Program
{
    static void Main(string[] args)
    {
        C c1 = new C();
        C c2 = new C();

        // 這一行需改為利用複製建構式的寫法
        // c2 = c1;
        c2 = new C(c1);

        // 要讓c1和c2的i分別等於10和20
        c1.I = 10;
        c2.I = 20;
    }
}

```

(d) (3%) 一個錯誤

```

class B
{
    private int i;
    public B(int i)
    {

```

```

        this.i = i;
    }
    public int GetData()
    {
        return i;
    }
}

```

```

class D : B
{
    private int j;
    public D(int i, int j) : base(i)
    {
        this.j = j;
    }

```

```

    public int GetData()

```

→ 成員函式與父類別成員函式同名，應在宣告時加上 **new** 關鍵字，或採用多型作法，此處宣告 **override**，而類別 **B** 之同名函式宣告為 **virtual**

```

    {
        return j;
    }
}

```

(e) (3%) 一組錯誤

```

class S
{
    private int i;
    public S(int i)
    {
        this.i = i;
    }
    abstract public int F() { return i; }

```

→ 抽象函式不能有實作主體，需將 **{ return i; }** 刪除，加上分號。又類別宣告也要加上 **abstract** 關鍵字

```

        // 必需是抽象函式
    }

```

3. 試寫出下列程式的輸出 (12%)

```
using System;

namespace Final2010Problem3
{
    class Employee
    {
        string id;
        string name;
        public Employee(string id, string name)
        {
            this.id = id;
            this.name = name;
            Console.WriteLine("Employee({0},{1})",
                id, name);
        }
    }
    class Faculty : Employee
    {
        string department;
        public Faculty(string id, string name,
            string department) : base(id, name)
        {
            this.department = department;
            Console.WriteLine("Faculty({0},{1}, {2})",
                id, name, department);
        }
    }
    class Teacher : Faculty
    {
        string course;
        public Teacher(string id, string name,
            string department, string course) :
            base(id, name, department)
        {
            this.course = course;
            Console.WriteLine("Teacher({0},{1},{2},{3})",
```

```

        id, name, department, course);
    }
}

class Program
{
    static void Main(string[] args)
    {
        Teacher skjeng = new Teacher("7777", "Jeng", "EE",
            "Computer Programming");
    }
}

```

Ans:



4. 試寫出下列程式的輸出 (12%)

```

using System;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;

```

```

namespace Final2010Problem4
{
    [Serializable]
    class B
    {
        private int i;
        public B(int i)
        {
            this.i = i;
        }
        public int I

```

```

        {
            get { return i; }
        }
    }

[Serializable]
class D : B
{
    private int j;
    public D(int i, int j) : base(i)
    {
        this.j = j;
    }
    public int J
    {
        get { return j; }
    }
}

class Program
{
    static void Main(string[] args)
    {
        B[] ba = new B[2];
        ba[0] = new D(3, 5);
        ba[1] = new B(1);
        BinaryFormatter formatter = new
            BinaryFormatter();
        try
        {
            FileStream output = new FileStream("Test.dat",
                FileMode.Create, FileAccess.Write);
            formatter.Serialize(output, ba[0]);
            formatter.Serialize(output, ba[1]);
            output.Close();
        }
        catch (SerializationException)
        {

```

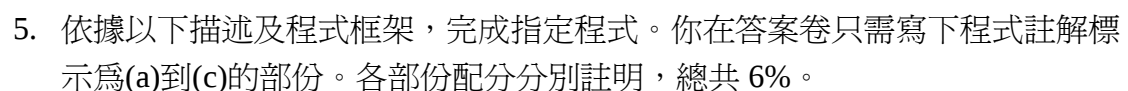
```

        Console.WriteLine(
            "Error writting to file in output");
    }
    catch (IOException)
    {
        Console.WriteLine(
            "Can not create or close file for output");
    }

    try
    {
        FileStream input = new
            FileStream("Test.dat",
                FileMode.Open, FileAccess.Read);
        try
        {
            int i;
            for(i=0; i<ba.Length; ++i)
            {
                Object obj =
                    formatter.Deserialize(input);
                if (obj.GetType() == ba[0].GetType())
                {
                    D di = (D)obj;
                    Console.WriteLine(di.I);
                    Console.WriteLine(di.J);
                }
                else if (obj.GetType() ==
                    ba[1].GetType())
                {
                    B bi = (B)obj;
                    Console.WriteLine(bi.I);
                }
                else
                {
                    throw new SerializationException();
                }
            }
        }
    }
}

```

Ans:



建立如下視窗介面，由使用者按鈕後，即在主選單顯示鬼牌。鬼牌之影像檔 J1.jpg 存放於專案 Final2010Problem5 之 bin 檔案夾內。鬼牌於主表單顯示之左上角座標為 5, 5；橫向及縱向大小分為 85 及 150 像素(pixels)。



```
// 檔案 MainForm.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Final2010Problem5
```

```

{
    public partial class MainForm : Form
    {
        /*******
        (a) 宣告必要物件變數 (1%)
        /*******

        public MainForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender,
            EventArgs e)
        {
            /*******
            DisplayImage();
            /*******
        }

        /*******
        private void DisplayImage()
        {
            // (b) 完成所需敘述 (4%)
        }
        protected override void
            OnPaint(System.Windows.Forms.PaintEventArgs e)
        {
            // (c) 完成所需敘述 (1%)
        }
        /*******
    }
}

```

Ans:

(a)

```

private Image image;
private Graphics graphics;
private bool started = false;

```

(b)

```
string fileName = "..\\J1.jpg";  
image = Image.FromFile(fileName);  
graphics = CreateGraphics();  
graphics.DrawImage(image, 5, 5, 85, 150);
```

(c)

```
base.OnPaint(e);  
if (started)  
{  
    DisplayImage();  
}
```

6. 撰寫一個 C# 程式以繪製歌曲之五線譜。為簡化問題，只要寫主控制台程式，並在螢幕依照旋律顯示各個音符之種類及音高即可。以台大校歌

國立臺灣大學校歌

沈剛伯詞

趙元任曲

5 3. 4 5 i | 2 i 7 6 5 | 5 3. 4 5 i | 3 i 7 6 5 | 5 5. 6 i 2 |

臺大的環境 野野 恩恩 臺大的氣象 物物 蓬蓬 遠望那玉山

3 3 2 i 2. 7 | 6 i 5 i 6 3 | 3 i 2 i - | 3 6 i i 7 | 6 5 6 0. 6 |

突出雲表 正象徵我們目標 的高峯 近看蜿蜒的淡水 他

3. 3 3 3 3 i | i 7 6 0. i | 6 i 5 i 5 i | 6 i. 2 3 | 2 - 0 i |

不搖擺地流動 正顯示我們百折不撓的作風 這

7 6 5 i. 7 | 6 i - 6 i | 5 i - 5 i | 3 5. i 2 | i - . 0 ||

百折不撓的作風 定使我們一切事業都成功

圖書館特藏組提供⁴

之第一小節為例，其輸出如下圖：

選取 C:\WINDOWS\system32\cmd.exe

QuarterNote: 65
DottedEighthNote: 62
SixteenthNote: 63
QuarterNote: 65
QuarterNote: 71
請按任意鍵繼續 . . .

表示其前五個音符分別為四分音符(quarter note)、附點八分音符(dotted eighth note)、十六分音符(sixteenth note)、四分音符、四分音符；而對應之音高以固定唱名音高之 MIDI 編號表示分為 65、62、63、65、71。在中央 C (Do)開始的一個八度內，音高與 MIDI 編號之關係如下表：

C (Do)	C#	D (Re)	D#	E (Mi)	F (Fa)	F#	G (Sol)	G#	A (La)	A#	B (Si)
60	61	62	63	64	65	66	67	68	69	70	71

以上述之台大校歌第一小節為測試場景撰寫程式，不需撰寫額外內容。不使用類別者，最高得 13 分；使用類別，不使用多型者，最高得 20 分；正確使用使用類別及多型者，最高得 25 分。

(25%)

Ans:

```
using System;

namespace Final2010Problem6
{
    interface MusicNote
    {
        void Draw();
    }
    class QuarterNote : MusicNote
    {
        private int pitch;
        public QuarterNote(int pitch)
        {
            this.pitch = pitch;
        }
        public void Draw()
        {
            Console.WriteLine("QuarterNote: {0}", pitch);
        }
    }
    class DottedEighthNote : MusicNote
    {
        private int pitch;
```

```

    public DottedEighthNote(int pitch)
    {
        this.pitch = pitch;
    }
    public void Draw()
    {
        Console.WriteLine("DottedEighthNote: {0}",
            pitch);
    }
}
class SixteenthNote : MusicNote
{
    private int pitch;
    public SixteenthNote(int pitch)
    {
        this.pitch = pitch;
    }
    public void Draw()
    {
        Console.WriteLine("SixteenthNote: {0}", pitch);
    }
}

class Program
{
    static void Main(string[] args)
    {
        MusicNote[] notes = new MusicNote[5];
        notes[0] = new QuarterNote(65);
        notes[1] = new DottedEighthNote(62);
        notes[2] = new SixteenthNote(63);
        notes[3] = new QuarterNote(65);
        notes[4] = new QuarterNote(71);
        DrawNoteList(notes);
    }

    static void DrawNoteList(MusicNote[] notes)
    {

```

```
    int i;  
    for (i = 0; i < notes.Length; ++i)  
    {  
        notes[i].Draw();  
    }  
}  
}
```