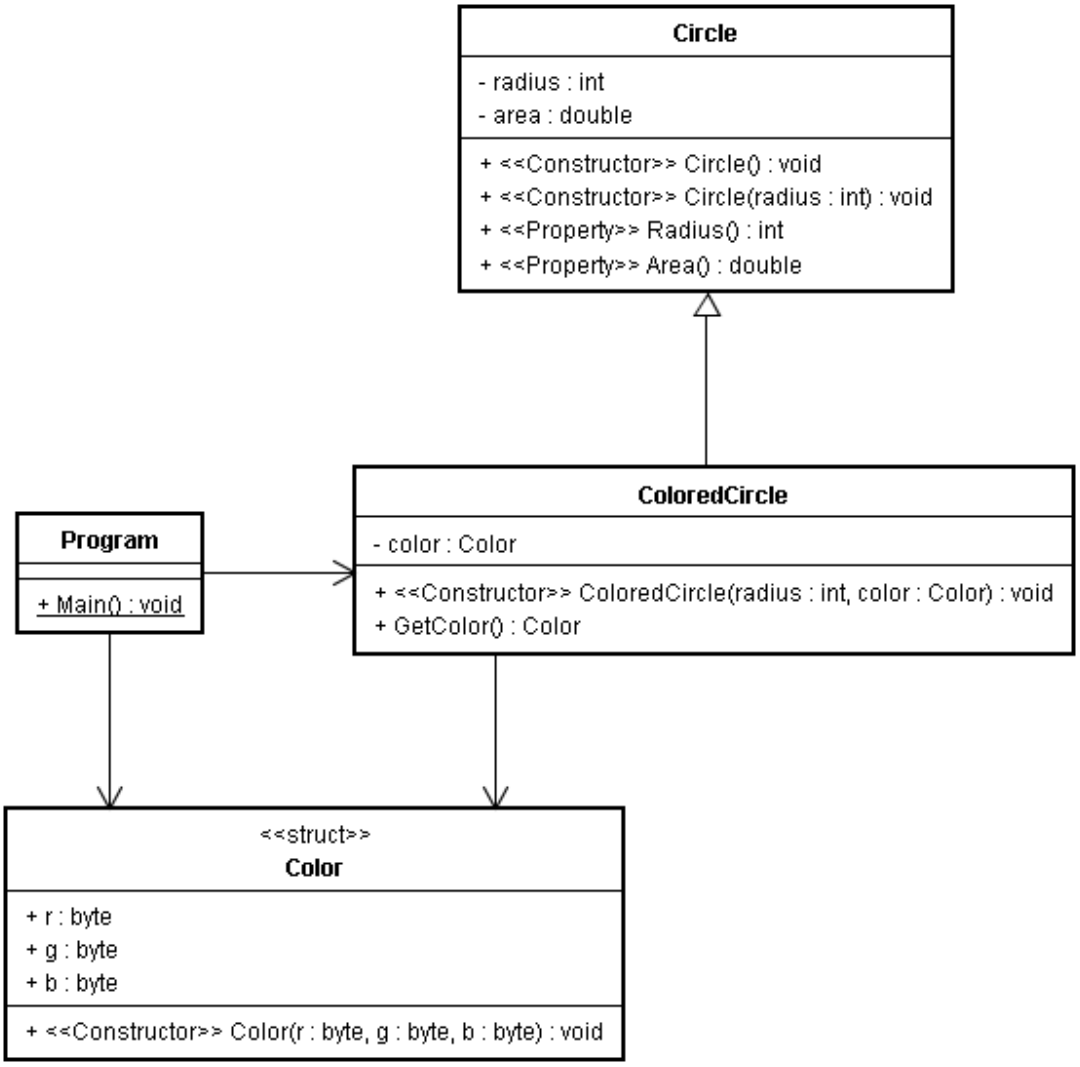
# 通識計算機程式設計期末考參考解答, 6/24/2011

1. 參考如下的 UML 類別圖，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 **`Circle`** 代表圓形，其中宣告私有 **`int`** 變數 **`radius`** 及 **`double`** 變數 **`area`**，分別代表半徑及面積；另有預設建構式設定 **`radius`** 爲 **`1`**，同時設定對應之 **`area`** 值；正規建構式輸入 **`radius`**，同時決定對應之 **`area`** 值，屬性 **`Radius`**、**`Area`** 分別傳回 **`radius`**、area 之值，此處不需處理例外情況 (6%)

Ans.

```
class Circle
 {
     private int radius;
     private double area;
```

```
    public Circle()
    {
        radius = 1;
        area = Math.PI*radius*radius;
    }
    public Circle(int radius)
    {
        this.radius = radius;
        area = Math.PI * radius * radius;
    }
    public int Radius
    {
        get { return radius; }
    }
    public double Area
    {
        get { return area; }
    }
}
```

(b) 撰寫結構 **Color**，包含三個公有 **byte** 變數 **r**、**g**、**b**，及正規建構式分別設定 **r**、**g**、**b** 之值 (3%)

Ans.

```
struct Color
{
    public byte r;
    public byte g;
    public byte b;
    public Color(byte r, byte g, byte b)
    {
        this.r = r;
        this.g = g;
        this.b = b;
    }
}
```

(c) 撰寫類別 **ColoredCircle** 繼承類別 **Circle**，並宣告私有 **Color** 變數 **color**，同時在宣告敘述中呼叫 **Color** 建構式 **Color(0, 0, 0)**設

定 **color** 之初值 (6%)

Ans.

```
class ColoredCircle : Circle
{
    private Color color = new Color(0, 0, 0);
}
```

(d) 撰寫類別 **ColoredCircle** 之正規建構式，記得呼叫父類別之建構式 (6%)

Ans.

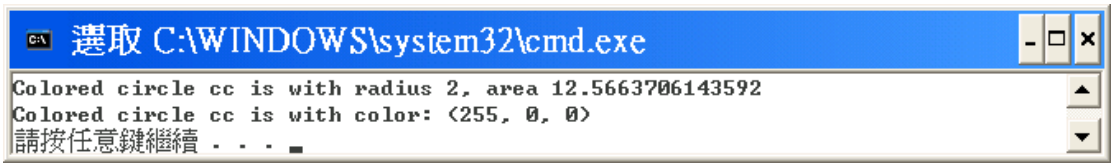
```
public ColoredCircle(int radius, Color c) :
    base(radius)
{
    color = c;
}
```

(e) 撰寫類別 **ColoredCircle** 之成員函式 **GetColor**，傳回 **color** (3%)

Ans.

```
public Color GetColor()
{
    return color;
}
```

(f) 寫一段測試主程式，設定顏色 **red** 之 **r**、**g**、**b** 爲(255, 0, 0)，**ColoredCircle** 物件之 **radius** 爲 **2**，**color** 爲 **red**，再利用它與繼承來的屬性、函式及結構之公開變數，輸出其半徑、面積、與顏色的 r、g、b 值如下圖 (6%)




Ans.

```
Color red = new Color(255, 0, 0);
ColoredCircle cc = new ColoredCircle(2, red);
Console.WriteLine("Colored circle cc is with " +
                  "radius {0}, area {1}",
                   cc.Radius, cc.Area);
Color color = cc.GetColor();
```

```
        Console.WriteLine("Colored circle cc is with "+
                          "color: ({0}, {1}, {2})",
                          color.r, color.g, color.b);
```

2. 找出以下程式片段之錯誤，並予更正.

(a) (3%) 一個錯誤

```
class A
{
    private int i;
    public A(int i)
    {
        this.i = i;
    }
    public static int Func()
    {
        return i;
    }
}
```

**Ans.**

靜態成員函式 **Func** 不應使用非靜態成員變數 **i**

改正：將成員函式 **Func** 改成非靜態

```
class A
{
    private int i;
    public A(int i)
    {
        this.i = i;
    }
    public int Func()
    {
        return i;
    }
}
```

(b) (3%) 一個錯誤

```
class A
```

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

    // 要把成員變數i乘以2
    public void Func(int i)
    {
        i *= 2;
    }
}
```

**Ans.**

成員函式 **Func** 中乘以 **2** 的 **i** 是輸入的型式參數，不是成員變數 **i**
所以成員變數 **i** 在 **Func** 中並未改變。
改正： 去掉成員函式 **Func** 的輸入參數

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

    public int I
    {
        get { return i; }
    }

    // 要把成員變數i乘以2
    public void Func()
    {
        i *= 2;
    }
}
```

(c) (3%).一項錯誤

```
class A
{
    private int i;
    public A(int i)
    {
        this.i = i;
    }
    public int I
    {
        get { return i; }
    }
}
class B : A
{
    private int j;
    public B(int i, int j)
        : base(i)
    {
        this.j = j;
    }
    public int J
    {
        get { return j; }
    }
    public void Func()
    {
        j = i;
    }
}
```

**Ans.** 類別 **B** 的成員函式 **Func** 不能使用父類別 **A** 的私有成員變數 **i**
改正: 將類別 **A** 中的成員變數 **i** 改宣告爲 **protecte**d，或
改寫類別 **B** 的成員函式 **Func** 內容爲 **j = I;**

```
class A
{
```

```
    protected int i;
    public A(int i)
    {
        this.i = i;
    }
    public int I
    {
        get { return i; }
    }
}

class B : A
{
    private int j;
    public B(int i, int j)
        : base(i)
    {
        this.j = j;
    }
    public int J
    {
        get { return j; }
    }
    public void Func()
    {
        j = i;
    }
}
```

或

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

```
    public int I
    {
        get { return i; }
    }
}

class B : A
{
    private int j;
    public B(int i, int j)
        : base(i)
    {
        this.j = j;
    }
    public int J
    {
        get { return j; }
    }
    public void Func()
    {
        j = I;
    }
}
```

(d) (3%) 一個錯誤

```
class A
{
    private int i;
    public A(int i)
    {
        this.i = i;
    }
    public virtual int Func()
    {
        return i;
    }
}
```

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

    // 要作爲多型之用
    public int Func()
    {
        return j;
    }
}
```

**Ans.** 子類別欲作爲多型之用的成員函式需加上 **override** 字樣
改正：Func 宣告時加上 override

```
class A
{
    private int i;
    public A(int i)
    {
        this.i = i;
    }
    public virtual int Func()
    {
        return i;
    }
}

class B : A
{
    private int j;
    public B(int i, int j)
        : base(i)
    {
```

```
        this.j = j;
    }

    // 要作爲多型之用
    public override int Func()
    {
        return j;
    }
}
```

(e) (3%) 一組錯誤

```
abstract class A
{
    public abstract int FunA();
}

abstract class B
{
    public abstract int FuncB();
}

class C : A, B
{
    int i;
    public C(int i)
    {
        this.i = i;
    }
    public int FuncA()
    {
        return i;
    }
    public int FuncB()
    {
        return i;
    }
}
```

**Ans.**

只有**interface**才可多重繼承，因此類別**B**不可同時繼承抽象類別**A**、**B**

改正：將抽象類別**A**、**B**改宣告爲**interfac**e，其中的函式去除public、abstract修飾詞

```
interface A
{
    int FuncA();
}

interface B
{
    int FuncB();
}

class C : A, B
{
    private int i;
    public C(int i)
    {
        this.i = i;
    }
    public int FuncA()
    {
        return i;
    }
    public int FuncB()
    {
        return i;
    }
}
```

3. 試寫出下列程式的輸出 (12%)

```
using System;

namespace Final2011Problem3
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            // 建立資料庫
            Book[] book = new Book[7];
            book[0] = new Book("紅樓夢", "曹雪芹", "  ",
                "好讀", "2007", "857.49 5514 2007b v1");
            book[1] = new Book("雪國 千鶴 古都", "川端康成",
                "高慧勤", "桂冠", "1994", "861.57 2200-10");
            book[2] = new Book("人鼠之間", "史坦貝克", "湯新楣",
                "金楓", "1987", "874.57 5046-2");
            book[3] = new Book("夢的解析", "佛洛伊德", "賴其萬",
                "志文", "1985", "175.1 2532");
            book[4] = new Book("國富論", "亞當.史密斯",
                "謝宗林","先覺","2000","550.1842 5034 2000");
            book[5] = new Book("天演論", "赫胥黎", "嚴復",
                "台灣商務", "1967", "143.45 4412 1967");
            book[6] = new Book("社會契約論", "盧梭", "何兆武",
                "唐山", "1987", "571.9 2143-75");
            DataBase db = new DataBase(book);

            // 資料搜尋
            db.ProcessQueryByAuthor("川端康成");
            db.ProcessQueryByTitle("社會契約論");
            db.ProcessQueryByCallNumber("143.45 4412 1967");

        }
    }

    class DataBase
    {
        private Book[] book;
        public DataBase()
        {
            book = null;
        }
```

```
public DataBase(Book[] book)
{
    this.book = book;
}
public void ProcessQueryByTitle(string title)
{
    for (int i = 0; i < book.Length; ++i)
    {
        if (book[i].Title == title)
        {
            book[i].DisplayData();
        }
    }
    Console.WriteLine();
}
public void ProcessQueryByAuthor(string author)
{
    for (int i = 0; i < book.Length; ++i)
    {
        if (book[i].Author == author)
        {
            book[i].DisplayData();

        }
    }
    Console.WriteLine();
}
public void ProcessQueryByCallNumber(string
    callNumber)
{
    for (int i = 0; i < book.Length; ++i)
    {
        if (book[i].CallNumber == callNumber)
        {
            book[i].DisplayData();

        }
    }
```

```
        Console.WriteLine();
    }
}

class Book
{
    private string title;
    private string author;
    private string translator;
    private string publisher;
    private string year;
    private string callNumber;
    public Book()
    {
        title = null;
        author = null;
        translator = null;
        publisher = null;
        year = null;
        callNumber = null;
    }
    public Book(string title, string author,
        string translator, string publisher,
        string year, string callNumber)
    {
        this.title = title;
        this.author = author;
        this.translator = translator;
        this.publisher = publisher;
        this.year = year;
        this.callNumber = callNumber;
    }
    public string Title
    {
        get{ return title; }
    }
    public string Author
    {
```

```
            get{ return author; }
        }
        public string CallNumber
        {
            get { return callNumber; }
        }
        public void DisplayData()
        {
            Console.WriteLine(title + "\t" +
                author + "\t" + translator + "\t" +
                publisher + "\t" + year + "\t" +
                callNumber);
        }
    }
}
```

Ans.


4. 試寫出以下程式在下列狀況時的輸出

```
// 程式 Final2011Problem4
using System;
using System.IO;

namespace Final2011Problem4
{
    class Program
    {
        static void Main(string[] args)
        {
            string fileName = "Test.dat";
```

```
    int answer = Test(fileName);
    Console.WriteLine("answer = " + answer);
}
static int Test(string fileName)
{
    int result = 0;
    try
    {
        StreamReader input = new
            StreamReader(fileName);
        Console.WriteLine("檔案開啓");

        try
        {
            int nDataPerLine =
                int.Parse(input.ReadLine());
            int i;
            string line;
            string[] dataString = new
                string[nDataPerLine];
            int data;
            int sum = 0;
            while (!input.EndOfStream)
            {
                line = input.ReadLine();
                dataString = line.Split(' ');
                for (i = 0; i < nDataPerLine; ++i)
                {
                    data = int.Parse(dataString[i]);
                    sum += data;
                }
            }
            result = 105 / sum;
        }
        catch (FormatException e)
        {
            Console.WriteLine("格式錯誤");
        }
```

```
                catch (DivideByZeroException e)
                {
                    Console.WriteLine("除以0例外");
                }
                catch (Exception e)
                {
                    Console.WriteLine(
                        "函式Test內層try-catch捕捉到例外");
                    Console.WriteLine(
                        "函式Test內層try-catch再拋出例外");
                    throw e;
                }
                finally
                {
                    input.Close();
                    Console.WriteLine("檔案關閉");
                }
            }
            catch(FileNotFoundException e)
            {
                Console.WriteLine("檔案不存在");
            }
            catch (Exception e)
            {
                Console.WriteLine(
                     "函式Test外層try-catch捕捉到例外");
            }
            return result;
        }
    }
}
```

(a) (3%) 檔案 Test.dat 尚未建立

Ans.

(b) (3%) 檔案 Test.dat 已在正確位置，且內容爲

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

**Ans.**


(c) (3%) 檔案 Test.dat 已在正確位置，且內容爲

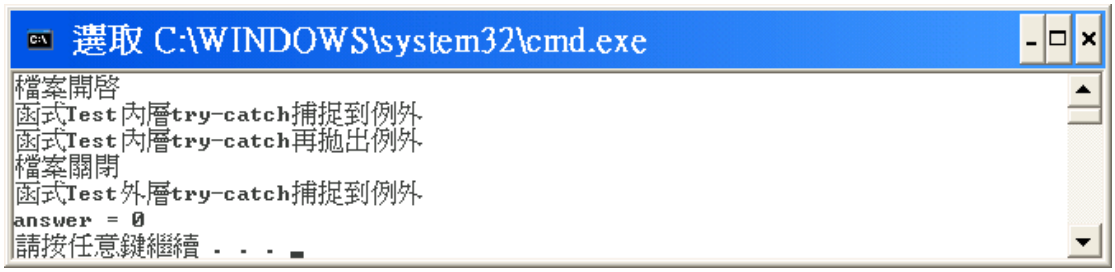
```
3
1 * 3
4 5 6
```

**Ans.**


(d) (3%) 檔案 Test.dat 已在正確位置，且內容爲

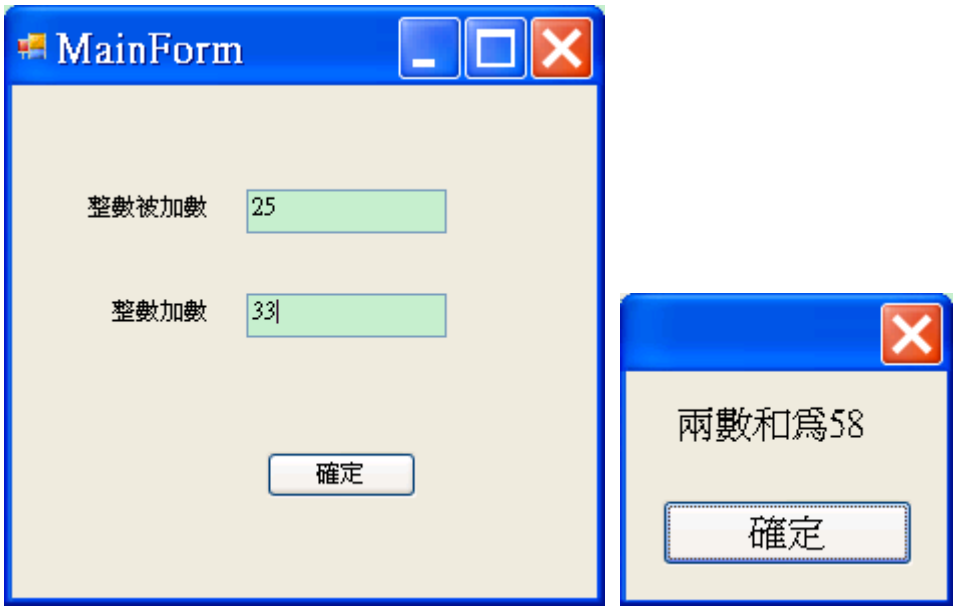
```
3
1 2 3
4 5
```

**Ans.**



5. 依據以下描述及程式框架，完成指定程式。你在答案卷只需寫下程式註解標示的部份。(6%)

程式描述:
建立如下視窗介面，由使用者輸入被加數及加數後，按下「確定」按鈕，即於對話盒顯示兩數之和。假設被加數及加數之文字盒分別由視窗設計工具自動設定爲 TextBox 物件 textBox1 及 textBox2，而「確定」按鈕則設爲 Button 物件 button1。




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

namespace Final2011Problem5
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }
```

```
        private void button1_Click(object sender, EventArgs e)
        {
            //*********************************************
            // 在此加入必要之程式碼
            //*********************************************
        }
    }
}
```

**Ans.**

```
            int x = Convert.ToInt32(textBox1.Text);
            int y = Convert.ToInt32(textBox2.Text);
            int sum = x + y;
            MessageBox.Show("兩數和爲"+sum.ToString());
```

6. 撰寫一個 C#程式以模擬神奇寶貝(Pocket Monster)運動會中的 30 米賽跑。爲簡化問題，只要寫主控台程式，並在螢幕顯示各個神奇寶貝的位置，最後顯示最先衝過終點最遠的贏家即可。假設參賽的神奇寶貝有皮卡丘(Pikachiu)

 、妙蛙種子(**Bulbasaur)**  、傑尼龜(**Squirtle)**  ，

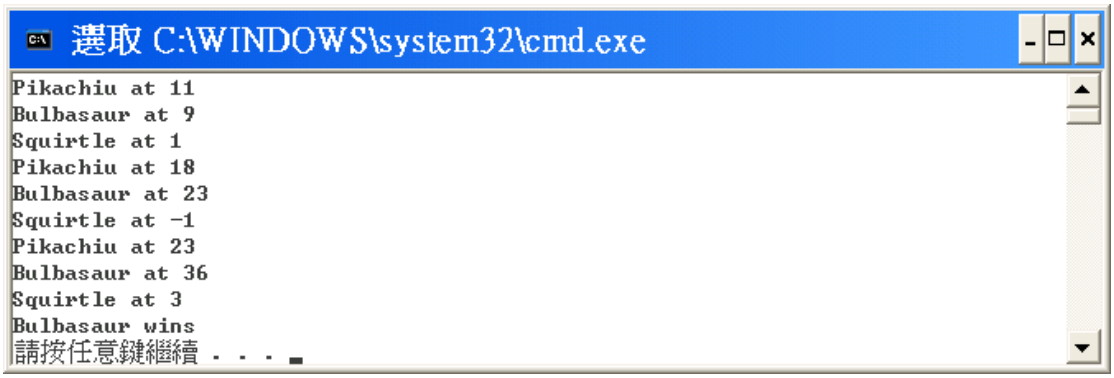
其每次移動的距離分別爲：

皮卡丘： **`rand.Next() % 16 + 1`**

妙蛙種子：**`(rand1.Next() % 10) + (rand2.Next() % 5) + 1`**
(需要兩個亂數產生器)

傑尼龜： **`(rand.Next() % 20 - 5) % 10  + 1`**
(沒錯，可能往反方向跑)

其輸出可能如下圖(殘念!! 皮卡丘先盛後衰，輸了！好討厭的感覺呀!)：




以上述之測試場景撰寫程式，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 Final2011Problem6
{
    class Program
    {
        static void Main(string[] args)
        {
            const int TRACK_LENGTH = 30;
            const int N_MONSTERS = 3;
            PocketMonster[] monsters = new
                PocketMonster[N_MONSTERS];
            monsters[0] = new Pikachiu(168);
            monsters[1] = new Bulbasaur(777, 545);
            monsters[2] = new Squirtle(66);

            bool finished = false;
            int i;
            while(!finished)
            {
                for(i=0; i<N_MONSTERS; ++i)
                {
                    monsters[i].Run();
```

```
            Console.WriteLine(monsters[i].Name +
                " at " + monsters[i].Position);
            if(monsters[i].Position >= TRACK_LENGTH)
            {
                finished = true;
            }
        }
        Console.WriteLine(
            "-------------------------------");
    }

    // check which monster runs farthest
    int maxLength = 0;
    for (i = 0; i < N_MONSTERS; ++i)
    {
        if (monsters[i].Position > maxLength)
        {
            maxLength = monsters[i].Position;
        }
    }

    // check for all winners (with possible tie
    // condition)
    for (i = 0; i < N_MONSTERS; ++i)
    {
        if (monsters[i].Position == maxLength)
        {
            Console.WriteLine(monsters[i].Name +
                " wins ");
        }
    }
  }
}

abstract class PocketMonster
{
  private string name;
  protected int position = 0;
```

```
    public PocketMonster(string name)
    {
        this.name = name;
    }
    public string Name
    {
        get { return name; }
    }
    public int Position
    {
        get { return position; }
    }
    abstract public void Run();
}

class Pikachiu : PocketMonster
{
    private Random rand;
    public Pikachiu(int seed)
        : base("Pikachiu")
    {
        rand = new Random(seed);
    }
    public override void Run()
    {
        position += rand.Next() % 16 + 1;
    }
}

class Bulbasaur : PocketMonster
{
    private Random rand1;
    private Random rand2;
    public Bulbasaur(int seed1, int seed2)
        : base("Bulbasaur")
    {
        rand1 = new Random(seed1);
        rand2 = new Random(seed2);
```

```
        }
        public override void Run()
        {
            position += (rand1.Next() % 10) +
                        (rand2.Next() % 5) + 1;
        }
    }

    class Squirtle : PocketMonster
    {
        private Random rand;
        public Squirtle(int seed)
            : base("Squirtle")
        {
            rand = new Random(seed);
        }
        public override void Run()
        {
            position += (rand.Next() % 20 - 5) % 10  + 1;
        }
    }
}
```

7. (額外加分題，至多 4%) 本課程中，你覺得：

甲、那一部份最有趣？爲什麼？

乙、那一部份最困難？爲什麼？

丙、講義有何可改進之處？

丁、怎麼做可以改進教學效果？