

類 科：資訊處理

科 目：程式設計概要

考試時間：1 小時 30 分

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。

一、請舉例說明(1)輸入方式限制(2)數值資料檢查(3)執行時期例外處理(4)資料存取權限控制等四種程式設計技巧，如何防止使用者輸入的資料，或程式執行過程產生的資料，發生資料值錯誤、不合理現象、執行錯誤程式中斷、或未符合資料限制等狀況。
(20 分)

二、如下程式，使用堆疊 (Stack) 的概念來存放數字基底轉換的結果，轉換完成後，再將 Stack 內的資料一一取出，作結果的顯示；請依程式中標號的說明，於程式中填入適當的程式碼，以完成十進位數字轉換為二、四、八、十六等四種進位制的數值。
(每小題 3 分，共 15 分)

(一)創造 ArrayList 物件

(二)設定迴圈起始狀態

(三)改變控制變數

(四)加入餘數 0~F 的字元到 ArrayList

(五)從最後加入的資料開始，一一取出存放在 ArrayList 中的資料 (ArrayList 資料的長度在 Java 中可使用 size()方法取得)

```
void base10toBaseNum(int sourceNum, int targetBase)
```

```
{ String ss = "";
  if (sourceNum < 0)
  { ss = ("轉換基底的數字必須大於等於零"); }
  else if ((targetBase == 2) || (targetBase == 4) ||
           (targetBase == 8) || (targetBase == 16))
  { char [] base16 = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };
    ArrayList result = /* (一) 創造 ArrayList 物件 */;
    for (int rr = /* (二) 設定迴圈起始狀態 */; rr > 0;
         rr = /* (三) 改變控制變數 */)
      { result.add(/* (四) 加入餘數 0~F 的字元到 ArrayList */); }
    ss = ("轉換" + sourceNum + "到基底" + targetBase + ": ");
    for(int i = /* (五) 由後往前取出 ArrayList 內容資料 */; i-- )
    { ss += (" " + result.get(i));
      if ((i%4) == 0) { ss += (" "); }
    }
    ss += "\n";
  }
  else { ss = ("轉換至的基底只能是 2, 4, 8, 16"); }
  /* print out ss */
} // End of base10toBaseNum
```

(請接第二頁)

類 科：資訊處理
科 目：程式設計概要

三、如下程式碼所構成的方法/函數（method/function），在呼叫時，使用 recursiveFlowControl(3, "ss", "dd", "aa")，請問該方法被呼叫執行後，顯示結果為何？如為語法有誤或執行時期發生中斷，請說明原因，並請修正錯誤或引起中斷的程式，讓程式可以正常執行結束並顯示結果。註：System.out.println()；會將小括號中的字串顯示到螢幕輸出裝置。（15分）

```
void recursiveFlowControl(int num, String s1, String s2, String s3)
{ System.out.println("... I am here ...");
  if (num == 1)
  { System.out.println("num==1, Move from " + s1 + " to " + s2); }
  else
  { recursiveFlowControl(num-1, s1, s3, s2);
    System.out.println("num!=1, Move from " + s1 + " to " + s2);
    recursiveFlowControl(num-1, s3, s2, s1);
  }
} // End of recursiveFlowControl
```

四、請使用 C-like 程式語言（C, C++, C#, Java...），撰寫一方法接受兩個已經由小到大排序好，但長度不等的整數型資料序列，回傳合併所傳入的兩個資料序列為一個由大到小排序好的新資料序列。限制：所撰寫的方法內，不可以使用所使用語言內建的排序方法輔助兩個排序好資料序列的合併。（20分）

五、如下的 HTML 程式執行後，執行頁面為何？請以圖示解釋執行結果。又當按下 "Submit" 動作按鈕後，會將帳號、密碼等資料，以不透過 URL 的方式，傳送給 login.php 程式來處理，請問應於表單標記中加入那些屬性，及相對應的資料值。（15分）

```
<html>
  <head> <title> Log in </title> </head>
  <body>
    <center><h1><strong>網際網路應用</strong></h1></center>
    <hr />
    <form id="Login" name="Login">
      <p> <label for="account"> Account </label>
        <input name="account" type="text" id="account" /> </p>
      <p> <label for="Password"> Password </label>
        <input name="Password" type="password" id="Password" /> </p>
      <p> <input type="reset" name="Reset" id="Reset" value="Reset" />
        <input type="submit" name="Submit" id="Submit" value="Submit" />
      </p>
    </form>
  </body>
</html>
```

（請接第三頁）

類 科：資訊處理
科 目：程式設計概要

六、就物件導向程式而言，現有如下的繼承關係的類別。（15分）

```
class GrandFather
{ String name = "GrandFather";
  String getName() { return name; }
  String whereIam() { return "class GrandFather"; }
} // End of GrandFather class
class Father extends GrandFather
{ String getName() { return name; }
  String whereIam() { return "class Father"; }
} // End of Father class
class Son extends Father
{ String name = "Son";
  String getName() { return name; }
  String whereIam() { return "class Son"; }
} // End of Son class
class Daughter extends Father
{ String getName() { return name; }
  String whereIam() { return "class Daughter"; }
} // End of Daughter class
```

於其他類別方法中使用如下的 statement：

```
Father f1 = new GrandFather();
Father f2 = new Son();
Father f3 = new Daughter();
System.out.println(f2.getName());
System.out.println(f3.getName());
System.out.println(f2.whereIam());
System.out.println(f3.whereIam());
```

請問編譯執行後，會顯示什麼結果？如為語法有誤或執行時期發生中斷，請說明原因，並將該 statement 予以忽視，然後說明可以正確執行 statement 所產生的結果。