

# 2014 協同產品設計實習報告

機械設計工程系二甲

April 23, 2014

## Contents

前言	4
<b>Pandoc 手冊</b>	<b>4</b>
協同產品設計實習專案 (2ag1) :	<b>5</b>
組員： . . . . .	5
w8 考試： . . . . .	5
w9 考試： . . . . .	6
w12： . . . . .	8
網際鼓式煞車設計 (2ag2)	8
程式設計架構	8
結果與討論	8
<b>W12 任務 (2ag3)</b>	<b>8</b>
摘要報告	9
網際鼓式煞車設計 (2ag4)	11
網際四連桿機構設計 (2ag5)	12
程式設計架構	12

結果與討論	12
網際鼓式煞車設計 (2ag6)	12
程式設計架構	12
結果與討論	12
網際鼓式煞車設計 (2ag7)	12
程式設計架構	13
結果與討論	13
網際四連桿機構設計 (2ag8)	13
程式設計架構	13
結果與討論	13
網際鼓式煞車設計 (2ag9)	13
程式設計架構	13
結果與討論	13
網際鼓式煞車設計 (2ag10)	14
程式設計架構	14
結果與討論	14
網際鼓式煞車設計 (2ag11)	14
程式設計架構	14
結果與討論	14
網際鼓式煞車設計 (2ag12)	14

程式設計架構	14
結果與討論	15
網際鼓式煞車設計 (2ag13)	15
程式設計架構	15
結果與討論	15
程式設計架構	15
結果與討論	15
網際鼓式煞車設計 (2ag15)	15
程式設計架構	15
結果與討論	16
網際鼓式煞車設計 (2ag16)	16
程式設計架構	16
結果與討論	16
第八週摘要報告	16
第九週摘要報告	17
網際鼓式煞車設計 (2ag18)	17
程式設計架構	17
結果與討論	17
網際鼓式煞車設計 (2ag21)	17
程式設計架構	17

結果與討論	18
網際 <b>OpenJSCAD</b> 程式設計 (coursemdetw)	18
設計程式架構	18
結果與討論	18

## 前言

協同產品設計實習課程目標<sup>1</sup>

Here is an inline note.<sup>2</sup>

學習協同產品設計流程與環境的基本原理與架構.

學習如何在網際專案管理系統的協助下, 進行協同產品設計.

學習如何在協同設計流程中, 進行有效率的工程設計表達與產品資料管理.

延續程式語言與電腦輔助設計實習課程, 學習如何建構協同產品設計環境所需的工具.

## Pandoc 手冊

<http://johnmacfarlane.net/pandoc/README.html>

footnotes

tables

flexible ordered lists

definition lists

fenced code blocks

superscript

subscript

strikeout

title blocks

automatic tables of contents

---

<sup>1</sup>這是註解的用法.

<sup>2</sup>Inlines notes are easier to write, since you don't have to pick an identifier and move down to type the note.

embedded LaTeX math

citations

markdown inside HTML block elements

## 協同產品設計實習專案 (2ag1) :

組員 :

40123101

40123102

40123132

OpenShift 網站: <http://2014cdag1-cadp13ag6.rhcloud.com/>

w8 考試 :

1. 請寫一個執行時可以列出 9×9 乘法表的網際 Python 程式, 然後 Push 到個人 bitbucket 空間, 而且同步指到 OpenShift 個人帳號上執行.

- 程式碼:

```
for x in range(0,10):  
    for y in range(0,10):  
        print(x, '*', y, '=', x*y)
```

- 解題過程:

打完程式後, 之後我把它上傳到 bitbucket, git add . → git commit -m "99" → git push, 之後為了要把 99 乘法表顯示在 openshift CMSimply 上, 進去 openshift CMSimply 的資料夾→ wsgi 資料夾→ application 檔案, 在裡面新增一個 class 的格式, 之後連線 FileZilla, 把 application 檔案覆蓋到 app-root/runtime/repo/wsgi 的 application 檔案下, 之後進去 openshift CMSimply 的網站, 就顯示 99 乘法表了。

- 解題心得:

雖然要打一個 99 乘法表的程式很快, 但要如何顯示在遠端的網頁上, 思考了很久, 在課堂後請教了 TA, 最後應用 cherrypy 的方式, 在 application 檔案, 新增一個 class 的格式, 就完成了。

2. 請將上述執行過程錄為 flv 後, 上傳到個人的 Vimeo 空間中, 並將網址回報到各組網站 (dokuwiki 與 CMSimply) 與報告中, 並且將相關心得與報告連結登錄到 wiki.mde.tw 第八週的分組頁面中.

- Bitbucket 連結: <https://bitbucket.org/40123102/40123102cmsimply/src>
  - Vimeo 空間: <https://vimeo.com/user26935042/videos>
  - copy 空間: <https://copy.com/XohJLdUBRJdk>
  - dokuwiki 網站: <https://40123102cdg1dokuwiki-cadp13ag6.rhcloud.com/doku.php?id=start>
  - CMSimply 網址: <http://40123102cdg1-cadp13ag6.rhcloud.com/w8test/>
3. 請在各組的雲端 dokuwiki 中, 根據下列 40 個帳號與密碼, 新增對應的使用者帳號與密碼後, 將雲端網址登錄在 wiki.mde.tw 各組第八週頁面中, 並說明操作過程與心得後, 將心得整理成 pdf 後繳交到 course@mde.tw.
- 解題過程:  
程式做不出來, 只好一個一個建立。
  - 解題心得:  
想很久, 還是無法想出程式, 只好用最勤勞的方法, 一個一個建立, 唉!!!

## w9 考試：

1. 請寫一個執行時可以列出以十為底對數表的網際 Python 程式, 然後 Push 到個人 bitbucket 空間, 而且同步指到 OpenShift 個人帳號上執行.
- 程式碼:

```
import math
count = 0
text='%04d'
for i in range(100,200+10):
    print(text%round(math.log(i/100,10)*10000,0),end='-')
    count += 1
    if(count == 10):
        print()
        count = 0
```
  - 解題過程:  
打完程式後, 有顯示出來, 但無法顯示表單在 CMSimply 網址上, 只能一個一個慢慢打。
  - 解題心得:  
想了很久, 還是無法, 只會打程式, 只好把顯示出來的數字一個一個加上, 慢慢打。

2. 請在個人的 OpenShift 平台上建立一個能夠列印出與九九乘法表結果完全相同的網際程式, 接著在乘法表上端加上兩個輸入表單, 讓使用者輸入兩個整數, 按下送出鍵後, 程式會列出以此兩個整數為基底的乘法表, 例如: 若兩個欄位都輸入: 9, 則列出九九乘法表, 若輸入 9, 20, 則列出  $9 \times 20$  的乘法表.

- 解題過程:

打完程式後, 之後我把它上傳到 bitbucket, `git add .` → `git commit -m "w9_2"` → `git push`, 之後為了要把 99 乘法表顯示在 openshift CMSimply 上, 進去 openshift CMSimply 的資料夾 → wsgi 資料夾 → application 檔案, 在裡面新增一個 class 的格式, 之後連線 FileZilla, 把 application 檔案覆蓋到 app-root/runtime/repo/wsgi 的 application 檔案下, 之後進去 openshift CMSimply 的網站, 輸入 9, 20, 則顯示  $9 \times 20$  的乘法表.

- 解題心得:

這題跟上週小考差不多, 只是還要再新增兩個輸入表單, 所以還要在回想一下, 但大致上 ok, 所以完成了。

3. 請在各組的雲端 dokuwiki 中, 新增帳號與密碼都是由 abc001 ~ abc399 字串所組成的 399 名用戶登入對應資料, 並將製作過程與驗證流程拍成 flv 後上傳到個人的 Vimeo 資料區, 並將連結放在個人第九週頁面.

- 程式碼:

```
import hashlib
# convert user_password into sha1 encoded string
def gen_password(user_password):
    return hashlib.sha1(user_password.encode("utf-8")).hexdigest()
text='abc%03d'
for i in range(1,399+1):
    print(text%(i)+':'+gen_password(text%(i))+':'+text%(i)+'@gmail.com:'+text%(i)+'@')
```

- 操作過程:

先寫一個程式使他能產生帳號與密碼都是由 abc001 ~ abc399 字串所組成的 399 名用戶登入對應資料, 之後連線到 openshift dokuwiki 的 FileZilla, 把產生出來的程式碼貼到 app-root/data/conf/users.auth.php 下, 之後開啟 dokuwiki 網站即可。

- 心得:

一開始這個程式需要思考一下, 如何使帳號與密碼都是由 abc001 ~ abc399 字串所組成的 399 名用戶, 打出來後還要去讓所產生的程式碼符合一開始的格式, 如同 users.auth.php, 思考很久最後終於成功了。

## w12 :

- 第十二週任務:

1. 請各組將第八週與第九週考試的摘要報告放入 Github 協同專案中的分組報告區, 並將內容放入各組控管的同步 OpenShift 網站中. (佔期末成績 5 分)
2. 請各組設法利用 CherryPy 與 Pro/Web.Link 技術, 在 Github 協同專案中建立一個能夠透過連結或表單控制 Cube 零件, a, b, 或 c 零件尺寸的網際協同程式, 讓使用者可以藉以利用近端的 Creo 嵌入式瀏覽器控制 Cube 的尺寸後列出該零件的體積大小. (佔期末成績 5 分)

- 心得:

1. 在 URL 直接更改參數: 在對應的程式內, 把迴圈改成可在 URL 後面輸入變數, EX:http://127.0.0.1:8080/cdag1/cube1?w=20&h=20&l=20
2. 在更改 FOR 迴圈時切記 {}! 在此 DEF 最下面還有一個括弧要刪除, 否則會跑出 CATCH

- 組員自評:

40123101 : 85 分

40123102 : 95 分

40123132 : 95 分

## 網際鼓式煞車設計 (2ag2)

有關鼓式煞車

### 程式設計架構

鼓式煞車

### 結果與討論

這裡是結果與討論

## W12 任務 (2ag3)

請各組將第九週考試的摘要報告放入 Github 協同專案中的分組報告區



## 摘要報告

第一題：

請寫一個執行時可以列出以十為底對數表的網際 Python 程式，然後 Push 到個人 bitbucket 空間，而且同步指到 OpenShift 個人帳號上執行。

```
import math
count = 0
text='%04d'
for i in range(100,200+10):
    print(text%round(math.log(i/100,10)*10000,0),end='-')
    count += 1
    if(count == 10):
        print()
        count = 0
```

解題心得：

可利用程式算出對數的值，就不需要一個一個按計算機。

第二題：

請在個人的 OpenShift 平台上建立一個能夠列印出與九九乘法表結果完全相同的網際程式，接著在乘法表上端加上兩個輸入表單，讓使用者輸入兩個整數，按下送出鍵後，程式會列出以此兩個整數為基底的乘法表，例如：若兩個欄位都輸入：9，則列出九九乘法表，若輸入 9，20，則列出 9×20 的乘法表。

```
class Example(object):
    _cp_config = {
        # if there is no utf-8 encoding, no Chinese input available
        'tools.encode.encoding': 'utf-8',
        'tools.sessions.on' : True,
        'tools.sessions.storage_type' : 'file',
        'tools.sessions.locking' : 'explicit',
        'tools.sessions.storage_path' : data_dir+'/tmp',
        # session timeout is 60 minutes
        'tools.sessions.timeout' : 60
    }

    @cherry.py.expose
    def index(self):
        output = ''
```

```

form = '''
<form action='action'>
num1:<INPUT type='text' name='num1'>
num2:<INPUT type='text' name='num2'>
<input type=submit>
<input type=reset>
</form>
'''

output += form
return output
@cherry.py.expose
def action(self, num1=9, num2=9):
    num1 = int(num1)
    num2 = int(num2)
    output = ''
    for i in range(num1):
        for j in range(num2):
            output += str(i) + '*' + str(j) + '=' + str(i*j) + '<br />'
    return output

```

解題心得：

利用網站輸入值並算出，並需要有兩個輸入表單。

第三題：

請在各組的雲端 dokuwiki 中，新增帳號與密碼都是由 abc001 ~ abc399 字串所組成的 399 名用戶登入對應資料，並將製作過程與驗證流程拍成 flv 後上傳到個人的 Vimeo 資料區，並將連結放在個人第九週頁面。

```

import hashlib
#convert user_password into sha1 encoded string
def gen_password(user_password):
    return hashlib.sha1(user_password.encode("utf-8")).hexdigest()
text='abc%03d'
for i in range(1,399+1):
    print(text%(i)+':'+gen_password(text%(i))+':'+text%(i)+'@gmail.com:'+text%(i)+'@gmail.com')

```

解題心得：

可以利用程式創造多人帳密，一起管理網站，減少一個一個創建帳密，又會有被盜用的風險。

```

import math count = 0 text='%04d' for i in range(100,200+10): print(text%round(math.log(i/100,10)*1000)) count += 1 if(count == 10): print() count = 0

```

```
class Example(object): cpconfig = { # if there is no utf-8 encoding, no
Chinese input available 'tools.encode.encoding': 'utf-8', 'tools.sessions.on' :
True, 'tools.sessions.storage_type' : 'file', 'tools.sessions.locking' : 'explicit',
'tools.sessions.storage_path' : data_dir+'/tmp', # session timeout is 60
minutes 'tools.sessions.timeout' : 60 }
```

```
@cherrypy.expose def index(self): output = "
```

```
form = '''
<form action='action'>
num1:<INPUT type='text' name='num1'>
num2:<INPUT type='text' name='num2'>
<input type=submit>
<input type=reset>
</form>
'''
output += form
return output
```

```
@cherrypy.expose def action(self, num1=9, num2=9): num1 = int(num1) num2
= int(num2) output = '' for i in range(num1): for j in range(num2): output
+= str(i) + '+' + str(j) + '=' + str(i*j) + ' ' return output
```

```
import hashlib #convert user_password into sha1 encoded string def
gen_password(user_password): return hashlib.sha1(user_password.encode("utf-
8")).hexdigest() text='abc%03d' for i in range(1,399+1): print(text%(i)+'.'+gen_password(text%(i))+'.'+

```

## 網際鼓式煞車設計 (2ag4)

github 連結 40123107: <https://github.com/40123107> 40123120: <https://github.com/40123120>  
cdag4: <https://github.com/2014cdag4/2014cdag4>

cmsimply 連結 40123107: <https://github.com/40123107> 40123120: <https://github.com/40123120>  
40123150: <http://cdg4-40123150.rhcloud.com/> ===

第九週 1.(第一題 30%) 請寫一個執行時可以列出以十為底對數表的網際 Python 程式, 然後 Push 到個人 bitbucket 空間, 而且同步指到 OpenShift 個人帳號上執行. [http://cdg4-40123150.rhcloud.com/get\\_page?heading=%E7%AC%AC%E4%B8%80%E9%A1](http://cdg4-40123150.rhcloud.com/get_page?heading=%E7%AC%AC%E4%B8%80%E9%A1)  
2.(第二題 40%) 請在個人的 OpenShift 平台上建立一個能夠列印出與九九乘法表結果完全相同的網際程式, 接著在乘法表上端加上兩個輸入表單, 讓使用者輸入兩個整數, 按下送出鍵後, 程式會列出以此兩個整數為基底的乘法表, 例如: 若兩個欄位都輸入: 9, 則列出九九乘法表, 若輸入 9, 20, 則列出 9×20 的乘法表. <http://cdg4-40123150.rhcloud.com/example/> <http://vimeo.com/92577008>  
3.(第三題 30%) 請在各組的雲端 dokuwiki 中, 新增帳號與密碼都是由 abc001 ~ abc399 字串所組成的 399 名用戶登入對應資料, 並將製作過程與驗證流程拍成

flv 後上傳到個人的 Vimeo 資料區, 並將連結放在個人第九週頁面. <https://php-40123150.rhcloud.com/doku.php?id=start> <http://vimeo.com/92577072> ===  
w12 請連結 <https://github.com/2014cdag4/2014cdag4> ===

這裡是結果與討論

## 網際四連桿機構設計 (2ag5)

有關連桿設計

### 程式設計架構

連桿計算

### 結果與討論

這裡是結果與討論

by 2014cdag5

## 網際鼓式煞車設計 (2ag6)

有關鼓式煞車

### 程式設計架構

鼓式煞車

### 結果與討論

這裡是結果與討論

## 網際鼓式煞車設計 (2ag7)

有關鼓式煞車

## 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論

## 網際四連桿機構設計 (2ag8)

有關連桿設計

## 程式設計架構

連桿計算

## 結果與討論

這裡是結果與討論

by 2014cdag5

## 網際鼓式煞車設計 (2ag9)

有關鼓式煞車

## 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論

這是第九組

## 網際鼓式煞車設計 (2ag10)

有關鼓式煞車

### 程式設計架構

鼓式煞車

### 結果與討論

這裡是結果與討論

## 網際鼓式煞車設計 (2ag11)

有關鼓式煞車

### 程式設計架構

鼓式煞車

### 結果與討論

這裡是結果與討論

## 網際鼓式煞車設計 (2ag12)

有關鼓式煞車

### 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論

## 網際鼓式煞車設計 (2ag13)

有關鼓式煞車

## 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論

<<<<<< HEAD 網際鼓式煞車設計 (2ag14) ===== 網際鼓式煞車設計 (2ag10) >>>>>> 26a5b9324111e15820203a5b96463a4196def80e ===

有關鼓式煞車

## 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論

## 網際鼓式煞車設計 (2ag15)

有關鼓式煞車

## 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論

## 網際鼓式煞車設計 (2ag16)

有關鼓式煞車

## 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論

## 第八週摘要報告

以協同方式寫 Wiki

本文目的在於讓學生利用開發的 wiki 與應用快速原型的協作式寫作的開發方法。而本文也由定性數據收集和分析方法的來評價。最後，對於協作方式來寫作的方式，其影響極其討論協議到組別與軟件開發方面的考慮和教學相關的要求問題研究方法 wiki 的協作式寫作，基於設計的研究包括以下四個步驟：1. 研究的現狀與認識目的並審查與 wiki 的協作式寫作相關的問題。2. wiki 將用於促進以協同方式寫作來設計，參與和群體互動。3. 使用多種方法收集其經驗數據。4. 通過系統的評價分析，並通過各種方法收集資料。分析，設計，實施和評價是相互共存的。而缺點提出於每個週期提出，重新設計，重新實現，並重新評估。維基應用域教育在協同方式寫作方法而成的 Wiki 是適用於多種情況，期望得到的 wiki 應用在學習投資效益與協同方式來寫作。它十分靈活的，足以適應的專業條件。更具體地，它的應用領域包括不同級別的更高教育，從研究生到社會教育。除了協同寫作與學科相關，維基可以為一些應用程序的開發系統來使用，如產生教材，網路上互相評論，並收集數據於一種項目。

結論和未來研究方向 1. 這項結果不能限制於狹小的研究的範圍，即使調查結果反映了那些報導中的一致性研究文獻。2. 用來判斷以不同方式來呈現協同，尤其是同組評議，這擁有極高的教育價值與分析性思維，並參與交流。可以促進團體間的合作和時間，wiki 打開了協同寫作上新的視野，而小組互動絕不是容易的事情，解決了技術，教學和文化各種問題。未來的工作將更顯的協同的重要性，wiki 以此設計為基礎下更加精進。重要的是，它更進一步影響高等教育對於 wiki 的使用。最後，它也進一步實踐於系統評論的基礎



## 第九週摘要報告

題目一: 請寫一個執行時可以列出以十為底對數表的網際 Python 程式, 然後 Push 到個人 bitbucket 空間, 而且同步指到 OpenShift 個人帳號上執行.

資料: <https://bitbucket.org/40123158/test>

題目三: 請在個人的 OpenShift 平台上建立一個能夠列印出與九九乘法表結果完全相同的網際程式, 接著在乘法表上端加上兩個輸入表單, 讓使用者輸入兩個整數, 按下送出鍵後, 程式會列出以此兩個整數為基底的乘法表, 例如: 若兩個欄位都輸入: 9, 則列出九九乘法表, 若輸入 9, 20, 則列出  $9 \times 20$  的乘法表.

資料: 第 17 組 dokuwiki , 新增 abc001 ~ abc399 用戶資料 <http://wikig17-weis.rhcloud.com/doku.php?id=start>

上傳影片 - 個人 Vimeo <https://vimeo.com/92577964>

Bitbucket 連結 <https://bitbucket.org/40123137/week9> Bitbucket 連結 <https://bitbucket.org/40123137/week9>

## 網際鼓式煞車設計 (2ag18)

有關鼓式煞車 g18

### 程式設計架構

鼓式煞車

### 結果與討論

這裡是結果與討論

## 網際鼓式煞車設計 (2ag21)

有關鼓式煞車

### 程式設計架構

鼓式煞車

## 結果與討論

這裡是結果與討論 00123

## 網際 OpenJSCAD 程式設計 (coursemdetw)

將 Spur 改為凸輪零件成型

## 設計程式架構

定義凸輪設計公式

## 結果與討論

有關凸輪程式設計的結果與討論