

# 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>	4
網際正齒輪減速機設計 (2ag1)	5
設計程式架構	5
結果與討論	5
網際鼓式煞車設計 (2ag2)	5
程式設計架構	5
結果與討論	5
<b>W12 任務 (2ag3)</b>	6
摘要報告	6
網際鼓式煞車設計 (2ag4)	8
網際四連桿機構設計 (2ag5)	9
程式設計架構	9
結果與討論	9

網際鼓式煞車設計 (2ag6)	9
程式設計架構	9
結果與討論	9
網際鼓式煞車設計 (2ag7)	10
程式設計架構	10
結果與討論	10
網際四連桿機構設計 (2ag8)	10
程式設計架構	10
結果與討論	10
網際鼓式煞車設計 (2ag9)	10
程式設計架構	10
結果與討論	11
第八週考試摘要	11
第八週報告	13
第九週考試摘要	14
第九週報告	15
第十二週	18
網際鼓式煞車設計 (2ag11)	19
程式設計架構	20
結果與討論	20

網際鼓式煞車設計 (2ag12)	20
程式設計架構	20
結果與討論	20
網際鼓式煞車設計 (2ag13)	20
程式設計架構	20
結果與討論	20
程式設計架構	21
結果與討論	21
網際鼓式煞車設計 (2ag15)	21
程式設計架構	21
結果與討論	21
網際鼓式煞車設計 (2ag16)	21
程式設計架構	21
結果與討論	21
第八週摘要報告	22
第九週摘要報告	22
網際鼓式煞車設計 (2ag18)	23
程式設計架構	23
結果與討論	23
網際鼓式煞車設計 (2ag21)	23

程式設計架構	23
結果與討論	23
網際 <b>OpenJSCAD</b> 程式設計 (coursemdetw)	23
設計程式架構	23
結果與討論	24

## 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

---

<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.

title blocks  
automatic tables of contents  
embedded LaTeX math  
citations  
markdown inside HTML block elements

## 網際正齒輪減速機設計 (2ag1)

正齒輪相關設計公式  
新增 commit 9 之後的資料

## 設計程式架構

傳遞功率  
新增 commit 9 之後的資料

## 結果與討論

這裡是結果與討論  
新增 commit 9 之後的資料

## 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.st\orage_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 }
```

```
@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) + ' ' 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  $9 \times 20$  的乘法表. <http://cdg4-40123150.rhcloud.com/example/> <http://vimeo.com/925770083>.(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

cd2ag10 報告 (2ag10) #####

以下為各週報告

## 第八週考試摘要

2014S CD Week8

課程教材:

A collaborative writing approach to wikis

Collabrative engine for distributed mechanical design

Web-based collaborative engineering support system

第八週考試題目

下列題目完成後, 必須 (1) 將程式碼送到個人的 Bitbucket repository 下 (2) 程式可在近端與雲端部署執行 (3) 在 wiki.mde.tw 個人第八週心得中留下 Bitbucket 與雲端執行連結 (4) 整理出一份小考第一 (二, 或三) 題的 PDF 檔案, 寄到 course@mde.tw, 標題為: 學號 -小考第一 (二, 或三) 題, 內容必須包含程式碼, 解題過程, 解題心得, Bitbucket 連結, 雲端網址等資料.

(第一題) 請寫一個執行時可以列出  $9 \times 9$  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.

(第二題) 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.

(第三題) (協同計分, 分組進行) 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. 40 個帳號與密碼

58B39 , 9J35UAVM

3624D , QANF34CW

7345B , 3PAFXKWZ

358DA , E6RJFKW4

3BC7B , HG2ASNYH

CA55C , XUZTHWQK

D2756 , UHK2W3D2  
368B8 , A5QGYA6W  
7948A , PF278WDQ  
C65C4 , GT4KBCXU  
A8964 , 3RPQSW2U  
22422 , 7E57K7F3  
9A5B4 , 4MVKRE5Z  
B5A76 , 3DTAAHUF  
57388 , NP39FGXR  
8A833 , C7DNBHCQ  
29AB8 , 6KMGK73Z  
5ABD2 , PV5FH722  
86293 , PJ69FBMS  
9DCBC , U5HR6QR8  
276DD , URE9FNWD  
599AA , X2P6CTXF  
9C449 , DKRN3V59  
64236 , 86UWN3E9  
43AAC , MNCJZCAX  
73B93 , QX945VJJ  
36283 , M3MQGUXD  
7447C , TQZVDKPT  
C73AB , MSP4GPPX  
8284B , XMT8W9RD  
62454 , SD4C7V89  
44B3C , 636DBRJC  
C75CB , M66RMMQ2  
4A3CB , FF485EQ4  
7D248 , FGJHQDAS  
A7CC4 , R47AHA4Y  
99BCC , RNQYZGQZ

6DA77 , HRCDP9D8

357AB , CKB4Q2EC

A9525 , JDVX75ST

第八週協同設計練習題目:

建立各組 OpenShift 上的 dokuwiki

<http://ethercalc.tw/> Openshift 上的 ethercalc

近端的 ethercalc

用 markdown 編輯電子書: <https://github.com/progit/progit> ,

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

期中考試與分組報告必選題

請以三組共 9 個人的情況下 (座位為 3×3 配置), 分析 2014s\_week3 的協同設計題目, 如何配置座位可以得到團隊的最大配分. 假如將組別擴大為五組共 15 人的情況下 (座位為 4×4 配置, 則團隊座位安排後的最大配合又是多少?

上課影片 <http://vimeo.com/user24079973/videos>

## 第八週報告

```
import cherrypy
class HelloWorld(object):
    @cherrypy.expose
    def index(self, var1=9, var2=9):
        # initialize outstring
        outstring = ""
        # initialize count
        count = 0
        d = int(var1)
        e = int(var2)+1
        for i in range(1, d):
            for j in range(1, e):
                count += 1
                #print(count)
                if count%(int(var2)) == 0:
                    outstring += "<td>"+str(i) + "*" + str(j) + "=" + str(i*j) + "</td>"+
                else:
                    outstring += "<td>"+str(i) + "x" + str(j) + "=" + str(i*j) + "</td>"+
            return "<table border=2><tr><td>99 乘法表</td></tr><tr>"+outstring
#http://127.0.0.1:8080/index?var1=10&var2=20
#"&nbsp;";"*4"
```

```
cherrypy.quickstart(HelloWorld())  
#application=cherrypy.Application(HelloWorld()) # 將符號打開上傳 openshift 即可使用，並刪除
```

以上程式碼即可跑出 99 乘法表.

9x9openshift: <http://9x9-cadp13ag8.rhcloud.com/> (因 openshift 空間不足，故利用第九週程式，程式碼略有差別，主要為多新增兩個輸入表單，其餘相同)

同步 bitbucket: <https://bitbucket.org/40123153/input9x9> (同上)

————我是分隔線————

影片 (由於網路超慢，怕錄製影片過大所以只錄製最後結果)

<https://vimeo.com/92003966>

P.S 由於題目理解錯誤，所以上述影片為失效。

## 第九週考試摘要

2014S CD Week9

課程教材:

A collaborative writing approach to wikis

Collabrative engine for distributed mechanical design

Web-based collaborative engineering support system

期中考試題目

下列題目完成後，必須 (1) 將程式碼送到個人的 Bitbucket repository 下 (2) 程式可在近端與雲端部署執行 (3) 在 [wiki.mde.tw](http://wiki.mde.tw) 個人第九週心得中留下 Bitbucket 與雲端執行連結以及其它參考連結 (4) 整理出一份期中考第一 (二, 或三) 題的 PDF 檔案, 寄到 [course@mde.tw](mailto:course@mde.tw), 標題為: cda\_學號\_姓名\_期中考第一 (二, 或三) 題 (乙班將 cda 改為 cdb), 內容必須包含程式碼, 解題過程, 解題心得, Bitbucket 連結, 雲端網址等資料.

(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.

(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.

(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.

第一題參考資料: 對數表與應用, 當  $x=10$ , 對應到 0 行的值, 表示要對 1.00 取以十為底的對數, 所得到的值為  $\text{math.log}(1.00, 10)=0$ , 而  $x=10$  對應到 1 行的值, 表示要對 1.01 取以十為底的對數, 所得到的值為  $\text{math.log}(1.01, 10)=0.004321373782642578$  然後再乘上 10000, 並且只取整數, 所以對應表的值為 43, 也就是表中的 0043, 當  $x=20$ , 而且對應到第 9 行的值, 則為  $\text{math.log}(2.09, 10)=0.32014628611105395$ , 然後再乘上 10000, 只取整數, 所以對應表上的值為 3201.

第一題參考: Python 中 `str()` 可以將整數或浮點數轉為字串, `int()` 則可以將字串轉為整數, `round(1.0123, 2)` 表示只取小數點後兩位, `math.log(x, 10)` 表示對  $x$  取以 10 為底的  $\log$  值, html 表格請參考.

期中成績評量

請各組依據 2014s\_week7 中的說明完成期中報告.(最後繳交期限為 2014.4.26 晚上 12:00)

成績評量時, 將依據各組在 [wiki.mde.tw](http://wiki.mde.tw) 中的頁面進行評量, 期中成績包括平時成績 (參考個人自評成績)、第八週考試與期中考試成績、期中報告成績等.

上課影片 <http://vimeo.com/user24079973/videos>

## 第九週報告

第一題:

Log 以 10 為底之表單 openshift: <http://log-cadp13ag8.rhcloud.com/>

bitbucket: <https://bitbucket.org/40123153/log10-1.00-1.99>

因為迴圈關係, 多了一個 21...還在更新中。(解決)

P.S 已利用 `elif` 判斷式將最後的 21 給剷除!

當在最後 20 的地方下 `elif` 將

給剷除, 故不在往下做表格及列印出 21 數字。

```
import cherrypy
import os
import math
# 1. 導入所需模組

# 2. 設定近端與遠端目錄
# 確定程式檔案所在目錄, 在 Windows 有最後的反斜線
_curdir = os.path.join(os.getcwd(), os.path.dirname(__file__))
# 設定在雲端與近端的資料儲存目錄
if 'OPENSIFT_REPO_DIR' in os.environ.keys():
    # 表示程式在雲端執行
```

```

download_root_dir = os.environ['OPENSIFT_DATA_DIR']
data_dir = os.environ['OPENSIFT_DATA_DIR']
else:
    # 表示程式在近端執行
    download_root_dir = _curdir + "/local_data/"
    data_dir = _curdir + "/local_data/"

# 3. 建立主物件
class HelloWorld(object):
    @cherry.py.expose
    def index2(self, input1=None, input2=None):
        return "Hello world!" + str(input1)
    @cherry.py.expose
    def inputform(self, input1=None, input2=None):
        return "input form" + str(input1)
    #index.exposed = True
    @cherry.py.expose
    def index(self):
        out=""
        c = 0
        k = 0
        for i in range(0,11):
            for j in range (0,10):
                k += 1
                c = 10 + i
                d = c/10+j/100
                e = (int(math.log(d,10)*10000))
                if (k%10 == 0 and k<=100):
                    out += "<td>" + str(e) + "</td></tr><td>" + str(c+1) + "</td><br />"
                elif k%100 == 0:
                    out += "<td>" + str(e) + "</td><br />"
                else:
                    out += "<td>" + str(e) + "</td>"
                #print("log=",int(math.log(d,10)*10000))
                #print("\n")
        return "<table border=1><tr><td>x</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td></tr><tr><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td></tr><tr><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td></tr><tr><td>40</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td></tr><tr><td>50</td><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td></tr><tr><td>60</td><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td></tr><tr><td>70</td><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td></tr><tr><td>80</td><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td></tr><tr><td>90</td><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td></tr></table>"

# 4. 安排啟動設定
# 配合程式檔案所在目錄設定靜態目錄或靜態檔案
application_conf = {'/static':{
    'tools.staticdir.on': True,
    'tools.staticdir.dir': _curdir+"/static"},
    '/downloads':{
    'tools.staticdir.on': True,
    'tools.staticdir.dir': data_dir+"/downloads"}
}
```



```

# 5. 在近端或遠端啟動程式
# 利用 HelloWorld() class 產生案例物件
root = HelloWorld()
# 假如在 os 環境變數中存在 'OPENSHIFT_REPO_DIR', 表示程式在 OpenShift 環境中執行
if 'OPENSHIFT_REPO_DIR' in os.environ.keys():
    # 雲端執行啟動
    application = cherrypy.Application(root, config = application_conf)
else:
    # 近端執行啟動
    '''
    cherrypy.server.socket_port = 8083
    cherrypy.server.socket_host = '127.0.0.1'
    '''
    cherrypy.quickstart(root, config = application_conf)

```

————我是分隔線————

第二題:

9×9 乘法表，並且有兩 input 可以改變 9×9 乘法表的範圍。例：input:9 and 20 則為 9×20 乘法表。

openshift: <http://9x9-cadp13ag8.rhcloud.com/> (備註:i 為前值，j 為後值)

bitbucket: <https://bitbucket.org/40123153/input9x9>

```

import cherrypy
class HelloWorld(object):
    @cherrypy.expose
    def index(self, var1=9, var2=9):
        # initialize outstring
        outstring = ""
        # initialize count
        count = 0
        d = int(var1)+1
        e = int(var2)+1
        for i in range(1, d):
            for j in range(1, e):
                count += 1
                if count%int(var2) == 0:
                    outstring += "<td>"+str(i) + "*" + str(j) + "=" + str(i*j) + "</td>"+
                else:
                    outstring += "<td>"+str(i) + "x" + str(j) + "=" + str(i*j) + "</td>"
        return "<table border=3><tr><td>99 乘法表</td></tr><tr>"+outstring+"''<br/><form>
        i 我是前值:<input type="text" name="var1"><br />
        j 我是後值:<input type="text" name="var2"><br />

```

```

        <input type="submit" value="send">
    </form>
    '''

#http://127.0.0.1:8080/index?var1=10&var2=20

#cherrypy.quickstart>HelloWorld()
application=cherrypy.Application>HelloWorld()

```

————我是分隔線————

第三題:

新增帳號與密碼都是由 abc001 ~ abc399 字串所組成的 399。

影片解說: <https://vimeo.com/92574166>

首先，我們必須要先有 abc001~abc399 的帳號密碼值，故先用程式迴圈跑出。  
(因為懶得寫輸出檔，或者是不熟悉所以不冒險去寫。)

將值複製至文件內將空白修掉，利用 EXCEL 把密碼在複製至 B 欄位。存成 CSV 檔

同樣利用文件將逗點修掉 (如果沒記錯空白處為一個 TAB，為了安全還是複製之前的。)

再利用程式，修改讀取的文件，將密碼處改為數列<sup>1</sup>

跑出 user 之後利用 FZ 將檔案覆蓋，則可以在雲端上登錄。(為了確認完全傳入，將 abc001 改為管理權限，確認帳號輸入)

dokuwiki\_openshift: <http://dokuwiki-cadp13ag8.rhcloud.com/doku.php>

bitbucket: [https://bitbucket.org/40123153/dokuwiki\\_abc](https://bitbucket.org/40123153/dokuwiki_abc)

第三題的程式碼實在不知道要打什麼….

## 第十二週

### 第十二週任務

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) github 小組倉儲 github 大分組倉儲

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) 1. Creo 必須使用 64 位元版本 (配合 Windows 7 操作系統) 2. Creo web\_enable\_javascript 設為 on, regen\_failure\_handling 設為 resolve\_mode 3. IE → 工具 → 網際網路選項 → 安

全性→信任的網站→自訂等級允許信任網站執行 ActiveX 4. cda 專案必須將 <http://cdag10-40123153.rhcloud.com/> 設為信任網站 5. 開啟 Creo 2.0, 建立一個 cube, relations:  $d0 = a$ ,  $d1 = b$ ,  $d2 = c$  (a, b, c 為 local parameters) 6. 然後在嵌入式 IE, 連接到 <http://cdag10-40123153.rhcloud.com/cdag10/cube10> (# 在 parameter 中將函式名稱改為 cube10) 7. 執行 <http://cdag10-40123153.rhcloud.com/cdag10/fourbar10> (# 在 parameter 中將函式名稱改為 fourbar10) 之前則需要先下載四連桿零件放在 V: 目錄中, 並開啟一個空的組立檔案, 執行時 Pro/Web.Link 程式會自動進行連桿組立. 8. Pro/Web.Link 應用: <http://www.ptc.com/company/news/inprint/taiwan/proe5.htm> 9. 參考資料: <http://inversionconsulting.blogspot.tw/search/label/WebLink>

首先必須將 creo 2.0 的 config.pro 檔做些設定, 必須要將 regen\_failure\_handling 設為 resolv\_mode, 使他能重繪。

再來必須將 web\_enable\_javascript 設為 on, 這樣才能讓 javascript 可以動作。

# 很重要!! 這將會成為成功與失敗的原因...

並且將 IE 關於 ActiveX 都通通啟用, 將 <http://cdag10-40123153.rhcloud.com/> 設為例外網頁。

再來就是繪圖, 將零件的三個尺寸設關係式,  $d0=a$ ,  $d1=b$ ,  $d2=c$ 。

所有前置動作都完成了, 再來就是利用 creo 裡面內建的 IE 跑囉!

將 <http://cdag10-40123153.rhcloud.com/cdag10/cube10> 貼至 IE 的 url 中, 就會開始跑囉!

(原始尺寸為 150x150x150 的方塊)

第一次:

第二次:

第三次:

第四次:

第五次:

..image..

大致上就這樣吧...不過 c 變數忘記動到...不過只是照 a 與 b 同樣的型式...將 a 或 b 複製下來改成 c 就可以三軸同動了。

終於結束了...在細節上有一點點小失誤就會整個失敗。

程式真的是很現實, 錯了就是錯了, 一個字打錯、一個步驟沒做到就是不行 (汗

## 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

有關凸倫程式設計的結果與討論