

# 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)	5
組員： . . . . .	5
w8 考試 . . . . .	5
w9 考試 . . . . .	6
w12 . . . . .	7
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
協同產品設計實習 (cdag6)	12
網際鼓式煞車設計 (2ag7)	13

程式設計架構	13
結果與討論	14
網際鼓式煞車設計 (2ag8)	14
程式碼	14
協同產品設計實習 (cdag9)	15
cd2ag10 報告 (2ag10)	16
第十組組員 (2ag10)	16
第八週考試摘要 (2ag10)	17
第八週報告 (2ag10)	19
第九週考試摘要 (2ag10)	20
第九週報告 (2ag10)	21
第十二週報告 (2ag10)	24
第十二週評分 (2ag10)	26
第十三週考試 (2ag10)	26
第十三週報告 (2ag10)	27
第十三週評分 (2ag10)	27
網際鼓式煞車設計 (2ag11)	28
程式設計架構	28
結果與討論	28
網際鼓式煞車設計 (2ag12)	28

程式設計架構	28
結果與討論	28
協同產品設計實習專案 (2ag13)	28
組員： . . . . .	28
協同產品設計實習專案 (2ag14)	29
網際鼓式煞車設計 (2ag15)	34
程式設計架構	34
結果與討論	34
網際鼓式煞車設計 (2ag16)	34
程式設計架構	34
結果與討論	35
第八週摘要報告	35
第九週摘要報告	35
網際 OpenJSCAD 程式設計 (coursemdetw)	37
設計程式架構	37
結果與討論	38

## 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>

---

<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.

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

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  $9 \times 9$  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.

- 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 9×20 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 9×20 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))+':'+text%(i)+'@gmail.com:'+text%(i)+'@gmail.com')
```

## 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

網際協同產品設計 (2ag5) 期末報告網站: <https://2014cdag5-yimin40123157.rhcloud.com>  
===

◎協同流程架構: 1. 說明: 善用各網站所提供的資源, 進行期末專案全班協同管理 2. 近端 USB/Portable hard drive: 儲存文件檔案資料 # Leo 程式管理, Pandoc 自動產生文件 3. 遠端 Openshift 網站: 免費最多能建立 3 個網頁 4. 遠端 Github 倉儲: 專案無限制人數協同, 免費倉儲只能 Public 5. 遠端 Bitbucket 倉儲: 免費專案最多 5 人協同, 倉儲能變更 Public/Privacy ===

◎ Pro/□ Web.Link: 1. 確認近端協同環境版次 2. Leo 4.11, Creo win7 64 位元 3. Creo Pro/□ Web.Link 透過 javascript 執行 #Creo only IE 4. 設定 Creo “File → Options → Configuration Editor” 底下 5. web\_enable\_javascript 設為 “on” 6. regen\_failure\_handling 設為 “resolve\_mode” 7. 設定 IE 瀏覽器 “網際網路選項” 底下 8.“安全性→□信任的網站→網站→新增 Pro/□ Web.Link 網站” 9.“自訂等級→ ActiveX 控制項與外掛程式” 所有選項設為 “●啟用” ===

◎ git 衝突解決: 1. git pull 2. git reset 3. git add -A 4. git status 5. git commit -m “fix problems” 6. git push

by 2014cdag5

## 協同產品設計實習 (cdag6)

組員：40123109 黃芯庭、40123119 呂冠緯、40123142 楊佳致  
小組 openshift 網站：<http://2014cdag6-40123109cd2014.rhcloud.com> 小組 github 網站：<https://github.com/2014cdag9/2014cdag6>

### === \* 第八週考試內容

(第一題) 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.

(解題過程) # 程式碼 for x in range(0,10): for y in range(0,10): print(x,'',y,'=',xy)

(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.

### \* 第九週考試內容

(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. (解題過程) # 程式碼 python 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 #openshift 連結 [https://2014s-40123109.rhcloud.com/get\\_page?heading=%C2%A0%E7%AC%AC%E4%B9%9D%E9%80%B1](https://2014s-40123109.rhcloud.com/get_page?heading=%C2%A0%E7%AC%AC%E4%B9%9D%E9%80%B1)

(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. (解題過程) #openshift 連結 <http://2014s-40123109.rhcloud.com/example/#bitbucket> 連結 <https://bitbucket.org/40123109/40123109cdg6>

(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. (解題過程) # 雲端 dokuwiki 頁面 <http://wiki-40123109.rhcloud.com/doku.php> #vimeo 連結 <https://vimeo.com/92575709>

\* 第十二週任務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

\* 第十三週任務任務一: 請以手動方式, 從下方下載 lego\_man.7z 零件, 完成組立後, 以組員為單位, 在各分組報告區詳細說明組立過程的重點與注意事項. (5分)

任務二：請利用 Vimeo 及 copy.com 存放任務一中的手動組立流程影片檔案，以組員為單位，將連結放入各分組報告區中。(5 分)

任務三：請仔細觀察協同程式中的 lego\_man 自動組立程式，並且從下方 lego\_parts\_edu.7z 中下載相同名稱的 lego\_man 零件，請詳細比較兩個壓縮檔案解開的 lego\_man 零件檔案，有哪些差別？請將詳細比對結果以組為單位將資料放入各分組的報告中。(5 分)

## 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

## 網際鼓式煞車設計 (2ag8)

第八組

組員：黃柏良王若嘉

第八週

(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

(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. 第九週

(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。(解題過程)

第一題

## 程式碼

```
python 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 #openshift 連結 https://2014s-40123109.rhcloud.com/get\_page?heading=%C2%
```

### 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  $9 \times 20$  的乘法表.

<http://cdg8-40123141.rhcloud.com/example/>

<https://bitbucket.org/40123141/>—

### 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.

<http://wiki-40123141.rhcloud.com/doku.php>

<http://vimeo.com/92589824>

### 第十二週

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

### 第十三週

任務一: 請以手動方式, 從下方下載 lego\_man.7z 零件, 完成組立後, 以組員為單位, 在各分組報告區詳細說明組立過程的重點與注意事項. (5 分)

任務二: 請利用 Vimeo 及 copy.com 存放任務一中的手動組立流程影片檔案, 以組員為單位, 將連結放入各分組報告區中. (5 分)

任務三: 請仔細觀察協同程式中的 lego\_man 自動組立程式, 並且從下方 lego\_parts\_edu.7z 中下載相同名稱的 lego\_man 零件, 請詳細比較兩個壓縮檔案解開的 lego\_man 零件檔案, 有哪些差別? 請將詳細比對結果以組為單位將資料放入各分組的報告中. (5 分)

## 協同產品設計實習 (cdag9)

組員: 40123114 王瑞豪組員: 40123117 吳昱緯組員: 40123127 洪宗佑

小組 openshift 網站: <https://2014cadg9-40123114.rhcloud.com/> 小組 github 網站: <https://github.com/coursemdetw/2014cda>

=== W8 考試

1. 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.

解題

// 設定一個 0 到 10 範圍,y 也設定 0 到 10, 接著印出 x,' ',y,'=',xy // 程式碼

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

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.
3. (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.

這題不太會做只能夠一個一個慢慢輸入

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. 解題

程式碼 `python 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` openshift 連結 <https://2014s-40123114.rhcloud.com/>

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 #openshift 連結 <http://2014s-40123114.rhcloud.com/example/> #bitbucket 連結 <https://bitbucket.org/40123114/40123114cdg9>

心得: 這題跟上一週的很相似, 只是變更了一下範圍, 又學習到了新的東西

(第三題 30%)3.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. 解題

<http://wiki-40123114.rhcloud.com/doku.php> #vimeo 連結 <https://vimeo.com/92575092>

心得: 第三題學習到如何用程式去輸入對應資料, 不必在一個一個利用手動方式再輸入了

W12 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

## cd2ag10 報告 (2ag10)

以下為各週報告

### 第十組組員 (2ag10)

小組網站:<http://goo.gl/aztMHi>

40123116-吳羽閔

40123118-吳聲麟

40123153-戴志軒

### 第八週考試摘要 (2ag10)

小組網站:<http://goo.gl/yi3qJ5>

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](http://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](mailto: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 Vemeo 空間中, 並將網址回報到各組網站 ([dokuwiki](http://dokuwiki.mde.tw) 與 [CMSimply](http://CMSimply.mde.tw)) 與報告中, 並且將相關心得與報告連結登錄到 [wiki.mde.tw](http://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 dowiki 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  $3 \times 3$  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  $4 \times 4$  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 (2ag10)

小組網站:<http://goo.gl/YCNd7p>

```
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

```

```

cherry.py.quickstart>HelloWorld())
#application=cherry.py.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 (2ag10)

小組網站:<http://goo.gl/8nPx2Y>

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 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, 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  $9 \times 20$  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 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 (2ag10)

小組網站:<http://goo.gl/Hy7Ktn>

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 給剔除！

當在最後 20 的地方下 elif 將

給剔除，故不在往下做表格及列印出 21 數字。

```
import cherryypy
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):
    @cherryypy.expose
    def index2(self, input1=None, input2=None):
        return "Hello world!" + str(input1)
    @cherryypy.expose
    def inputform(self, input1=None, input2=None):
        return "input form" + str(input1)
    #index.exposed = True
    @cherryypy.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><t

```

# 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):

```

```

@cherry.py.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

#cherry.py.quickstart(HelloWorld())
application=cherry.py.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….

## 第十二週報告 (2ag10)

小組網站:<http://goo.gl/28PW52>

### 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:<https://github.com/2014cdag10>

github 大分組倉儲:<https://github.com/coursemdetw/2014cda>

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:3375000

第二次:4096000

第三次:4913000

第四次:5832000

第五次:6859000

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。

## 第十二週評分 (2ag10)

40123116 : 10 分

40123118 : 10 分

40123153 : 10 分

## 第十三週考試 (2ag10)

小組網站:<http://goo.gl/til1Nq>

<https://vimeo.com/95823422> copy.com: <https://copy.com/P9fGFaLSlQqx>

<https://vimeo.com/95823554> copy.com: <https://copy.com/1pnPvVduOJ9B>

任務一：請以手動方式，從下方下載 lego\_man.7z 零件，完成組立後，以組員為單位，在各分組報告區詳細說明組立過程的重點與注意事項。(5 分)

任務二：請利用 Vimeo 及 copy.com 存放任務一中的手動組立流程影片檔案，以組員為單位，將連結放入各分組報告區中。(5 分)

任務三：請仔細觀察協同程式中的 lego\_man 自動組立程式，並且從下方 lego\_parts\_edu.7z 中下載相同名稱的 lego\_man 零件，請詳細比較兩個壓縮檔案解開的 lego\_man 零件檔案，有哪些差別？請將詳細比對結果以組為單位將資料放入各分組的報告中。(5 分)

主要任務：了解各組員對於課程內容的了解 (採每一組員分別完成任務進行測試)

協同環境與架構

實際操作

Creo 零件手動組立

Creo 零件參數手動設定

Creo 零件參數網際變更

Creo 零件自動組立

lego\_man.7z

lego\_parts\_edu.7z

weblink user guide.pdf

## 第十三週報告 (2ag10)

小組網站:<http://goo.gl/QFJp5F>

因為只有說利用 creo，但並沒限定使用 Parametric 或 Direct. 所以利用 Direct 必較人性化…所以用 Direct 組立 (主要是沒有順序問題!)

首先載入身體，放置在任意座標.

在載入左、右手臂，利用同心圓及重合將其組上.(注意面的位子，將要組的面先轉至要組上的面的方向)

之後將手載入兩根，利用同心圓及重合組在手臂前端.

將下盤組在身體上，利用三個平面的重合即可.

完成後將左、右腳利用同心圓及重合組在下盤兩端.(注意面的位子，將要組的面先轉至要組上的面的方向)

載入頭部，同樣利用同心圓及重合組在身體上 (注意孔大小)

載入帽子，利用三個重合即可 (第一個是曲面的重合，中間平面的重合、下平面的重合)

小組 vedio: <https://vimeo.com/95822338>

小組 copy.com: <http://goo.gl/K1nh1G>

lego\_man 自動組立程式組出來的 Feat ID 是順序的 40 41 42 …一個一個零件載入裝配. 而 lego\_parts\_edu.7z 中下載相同名稱的 lego\_man 零件卻不是這樣順序的… 這代表他並不是一次就組起來，可能途中有改變、刪除什麼的.

自動組立的某些限制條件或許沒訂得很好，動作有點好笑！

P.S 自動組立不資源 creo 2.0 M70(x64) 版本，會跑出錯誤！只可以用於 M70(x64) 版本

P.S2 無法將 man.py 程式移植到我們 (cdag10) 小組網頁中，也就是說無法利用 <https://cdag10-40123153.rhcloud.com/cdag10/man/>開啟…更無法利用 <https://cdag10-40123153.rhcloud.com/cdag10/man/assembly> 來跑自動組立.

## 第十三週評分 (2ag10)

40123116 : 15 分

40123118 : 15 分

40123153 : 15 分

## 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

## 協同產品設計實習專案 (2ag13)

組員：

40123155

40123124

40123138

[illegible]

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)

考試摘要 <http://dokuwiki-40123155.rhcloud.com/doku.php>

- bitbucket 網站: <https://40123155@bitbucket.org/40123155/—.git>
- dokuwiki 網站: <http://dokuwiki-40123155.rhcloud.com>
- vimeo 網站: <https://vimeo.com/user26959367>
- CMSimply 網站: <https://python-40123124.rhcloud.com>
- 下載網站: <https://copy.com/SjKQ8LHZpVFq>
- HTML 網站: <http://dokuwiki-40123155.rhcloud.com/lib/exe/fetch.php?media=%E6%9C%9F%E4%>

<http://dokuwiki-40123155.rhcloud.com/lib/exe/fetch.php?media=%E8%AA%B2%E7%A8%8B%E6%95%>

組員自評:

40123155 : 80 分

40123124 : 70 分

40123138 : 75 分

## 協同產品設計實習專案 (2ag14)

w8

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](http://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](mailto: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

“”

$$1*1=1$$

$$1*2=2$$

$$1*3=3$$

$$1*4=4$$

$$1*5=5$$

$$1*6=6$$

$$1*7=7$$

$$1*8=8$$

$$1*9=9$$

$$2*1=2$$

$$2*2=4$$

$$2*3=6$$

$$2*4=8$$

$$2*5=10$$

$$2*6=12$$

$$2*7=14$$

$$2*8=16$$

$$2*9=18$$

$$3*1=3$$

$$3*2=6$$

$$3*3=9$$

$$3*4=12$$

$$3*5=15$$

$$3*6=18$$

$$3*7=21$$

$$3*8=24$$

$$3*9=27$$

$$4*1=4$$

$$4*2=8$$

$$4*3=12$$

$$4*4=16$$

$$4*5=20$$

$$4*6=24$$

$$4*7=28$$

$$4*8=32$$

$$4*9=36$$

$$5*1=5$$

$$5*2=10$$

$$5*3=15$$

$$5*4=20$$

$$5*5=25$$

$$5*6=30$$

$$5*7=35$$

$$5*8=40$$

$$5*9=45$$

$$6*1=6$$

$$6*2=12$$

$$6*3=18$$

$$6*4=24$$

$$6*5=30$$

$$6*6=36$$

$$6*7=42$$

$$6*8=48$$

$$6*9=54$$

$$7*1=7$$

$$7*2=14$$

$$7*3=21$$

$$7*4=28$$

$$7*5=35$$

$$7*6=42$$

$$7*7=49$$

$$7*8=56$$

$$7*9=63$$

$$8*1=8$$

$$8*2=16$$

$$8*3=24$$

$$8*4=32$$

$$8*5=40$$

$$8*6=48$$

$$8*7=56$$

$$8*8=64$$

$$8*9=72$$

$$9*1=9$$

$$9*2=18$$

$$9*3=27$$

$$9*4=36$$

$$9*5=45$$

$$9*6=54$$

$$9*7=63$$

$$9*8=72$$

$$9*9=81$$

“”

(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.

“” import hashlib

“” sha1 secure hashes “” st\_date = []

```
def gen_password(user_password): return hashlib.sha1(user_password.encode("utf-8")).hexdigest()
```

```
with open("test.txt", "wt") as out_file: with open("2014.txt", "r", encoding="utf-8") as in_file: for index in in_file:
```

```
    number, name = index.strip().split(' ')
    mail = number + "@gm.nfu.edu.tw"
    print(number, name)
    st_date.append([number, name, mail])
    out_file.write(str(number) + ":" + gen_password(str(number)) + ":" + str(mail) + ":user" + "\n")
```

```
“”
```

```
==== w9
```

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 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, 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.

程式: <https://bitbucket.org/40123136/444>

(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.

程式: <https://bitbucket.org/40123136/99>

(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.

Vimeo : <https://vimeo.com/92577999>

Bitbucket

<https://bitbucket.org/40123136/444> <https://bitbucket.org/40123136/99>

OpenShift

<http://cda14-40123136.rhcloud.com/> <http://python-cadpag5.rhcloud.com/>  
<https://2014a14-40123136.rhcloud.com/>



Vimeo

<https://vimeo.com/92577999>

=== w13

任務一：請以手動方式，從下方下載 lego\_man.7z 零件，完成組立後，以組員為單位，在各分組報告區詳細說明組立過程的重點與注意事項。(5 分)

1. 開啟 lego\_body.prt(主體) 之後接上 lego\_arm\_rt.prt、lego\_arm\_lt.prt 2. 加上 lego\_hand\_lt.prt 再來就將機器人的頭加上 lego\_head.prt 3. 在接上 lego\_waist.prt 再來就將雙腿接上 4. 最後將 lego\_head.prt 及 lego\_hat.prt 機器人組合就完成了，之後利用爆炸圖可以得到爆炸後的各零件但需要再調整一下位置，就能完成良好的爆炸圖即可完成作業這是一個組合圖教學珍貴的經驗，一個了解自動組合程式的機會。

任務二：請利用 Vimeo 及 copy.com 存放任務一中的手動組立流程影片檔案，以組員為單位，將連結放入各分組報告區中。(5 分)

小組 vedio: <https://vimeo.com/95872800>

小組 copy.com: <https://copy.com/kJl91O34CfvA>

任務三：請仔細觀察協同程式中的 lego\_man 自動組立程式，並且從下方 lego\_parts\_edu.7z 中下載相同名稱的 lego\_man 零件，請詳細比較兩個壓縮檔案解開的 lego\_man 零件檔案，有哪些差別？請將詳細比對結果以組為單位將資料放入各分組的報告中。(5 分)

lego\_man 自動組立程式組出來的 Feat ID 是 40 41 42 的零件載入裝配。lego\_parts\_edu.7z 中下載的 lego\_man 零件卻不是這樣順序的… 這代表他不是一次就組起來的

===

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>

協同產品設計實習專案 (2ag18)

=====

組員:

40123126 柯嘉洋

40123128 胡瀚中

40123149 蔡東維

=====

OpenShift:

=====

第十三週任務:

任務一:

40123149:

步驟: 建立新的組立檔, 後依序步驟逐漸組裝, 而插入 Lego 零件的順序有影響,

如果胡亂插入會有無法組裝的問題, 所以要特別注意, 而每次插入零件檔時要先調整好位置並藉由基準面與基準面對齊, 便可完成。

解題步驟圖片: <https://copy.com/gHzVnqHZnpZt>

心得: 經過這次的試題後, 組裝零件要考慮的事情也是很多的。

2ag21

組員: 40023143、40123143、40123145、40123156

本組 CMSimply 網站: <http://cmsimply-40123156.rhcloud.com/>

=== 第十三週

任務一: 請以手動方式, 從下方下載 lego\_man.7z 零件, 完成組立後, 以組員為單位, 在各分組報告區詳細說明組立過程的重點與注意事項. (5 分)

任務二: 請利用 Vimeo 及 copy.com 存放任務一中的手動組立流程影片檔案, 以組員為單位, 將連結放入各分組報告區中. (5 分)

任務三: 請仔細觀察協同程式中的 lego\_man 自動組立程式, 並且從下方 lego\_parts\_edu.7z 中下載相同名稱的 lego\_man 零件, 請詳細比較兩個壓縮檔案解開的 lego\_man 零件檔案, 有哪些差別? 請將詳細比對結果以組為單位將資料放入各分組的報告中. (5 分)

=== 第十三週 (40023143)

任務一: 1. 開啟 lego\_body.prt 利用重合將開啟後的 lego\_arm\_rt.prt 和 lego\_arm\_lt.prt 接上, 再改變角度。2. 開啟 lego\_hand\_lt.prt 接上 lego\_arm\_rt.prt 和 lego\_arm\_lt.prt 完成身體與手 3. 開啟 lego\_head.prt 接在身體上在開啟 lego\_hat.prt 完成頭的部分 4. 開啟 lego\_waist.prt 接在身體下。5. 最後開啟 lego\_leg.prt 和 lego\_waist.prt 接合, 完成 lego\_man 6. 利用爆炸圖的功能將 lego\_man 爆炸。心得: 利用 creo 將各個零件組合起來, 真的很有趣。但因為操作不順和錄製程式的畫質調太低, 影片不太成功, 但是確實體會了 creo 的方便, 利用雲端將畫好的檔案下載, 再利用 creo 將複雜的零件組合, 真的可以大大提升工作效率。

任務二: Vimeo: <https://vimeo.com/95867694> copy.com: <https://www.copy.com/s/DEGh0OTCbKDD/333>

任務三: lego\_man 利用自動組立程式組出來的 Feat ID 的排序是 40、41、42、43……以此類推, 一個一個零件檔載入裝配; 而 lego\_parts\_edu.7z 中下載相同名稱的 lego\_man 零件檔卻不是這樣排序的, 表是 lego\_man 不是一次就組起來, 途中應該有改變、刪除……等程序。

=== 第十三週 (40123143)

任務一: 首先, 先開啟 lego\_body.prt(主體) 之後接上 lego\_arm\_rt.prt、lego\_arm\_lt.prt 之後再加上 lego\_hand\_lt.prt 再來就將機器人的頭加上 lego\_head.prt, 在接上 lego\_waist.prt 再來就將雙腿接上最後將 lego\_head.prt 及 lego\_hat.prt, 機器人組合就完成了, 最後利用爆炸圖可以得到爆炸後的各零件但需要在調整一下位置, 就能完成良好爆炸圖即可完成作業, 謝謝老師的教導, 這是一個很特別的組合圖教學, 希望能快點了解自動組立程式。

=== 第十三週 (40123156)

任務一: 組立過程為 1. 開啟 creo 2. 新建“組立” 3. 用“插入”功能加入零件 4. 設定零件間的限制條件注意事項為設定零件間的限制條件要盡量設定到零件完全不能移動

任務二: Vimeo: <https://vimeo.com/95857245> copy.com: <https://copy.com/zJSJ6ZVGdtsy>

任務三: lego\_parts\_edu.7z 裡的樂高零件比較多可是組合起來卻沒有太大差異，第二個壓縮檔裡的零件可能是分成更細小的零件。

## 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