Software development for Business 2 (BIF-4-SD2)

Assessment 2013/2014

Draft

To be submitted by

4pm on Thursday 15th May 2014 to L105

This assessment forms the central focus of this module. It is where your will get your marks, but it is also the main learning vehicle. You will be developing an increasingly more complex piece of software, each stage having marks attached to it. You will provide evidence that you have been able to do the various parts of this assessment.

Required:

You are supplied with 3 data files and a set of questions. These are yours and are different from everyone else. Your work must be done using your questions and your data files. The example programs and the lectures will support you in learning how to answer your questions.

Stages:

1. Write Python code to answer your questions. The user should be able to enter the file name and the program should then display the answers to your questions. (Preferably only reading the whole file once).
2. The program should then be modified to follow good practice, which means that the program be altered to first be structured using functions (*Stage 1*) and then later to use classes and objects correctly, (*Stage 2*) and be well documented.
3. The next development is to make the program work using the GUI (*Stage 3)* to allow the user to enter the file name and to display the answers.
4. You will select one research topic from a supplied list and
   (a) Write a summary of your findings and
   (b) Apply and demonstrate your findings to one of the stages you have been developing.
5. You need to write an evaluation of your achievements.
6. You should demonstrate that the program is correct by the use of unit tests.

Deliverables:

Your submission should have:

a. The pages numbered, and a table of contents after the Title Page.
b. The title page containing at the very least, the module name, your name, your tutors name and your student number.
c. Your source code (annotated) from each stage of your development.
d. Evidence of your program running at each stage, with written commentary (*that can be hand written)*
e. Your research report.
f. An evaluation of what you have achieved and the problems you found.
g. Your work held in a flat binder or plastic sleeve. (Not ring bound)

Submission: You need to print out and present the following:

a. **The program code at each stage** (Plain command line, using Objects, GUI and tests). This should be annotated, either by the use of comments or by writing by hand on the printouts. If you must use a word processor, please make sure that the program code is in Courier font, reduced in size to 10pt and has a reasonably small tab size. Each time you move onto the next stage, you need to keep a copy of the original work you have done up to that point. Please put the evidence of the program running immediately after the printout of the program source code, then any testing code and output from the tests.

b. **Evidence of your program** running for each stage you have completed. You should be aware that you do not have to complete all of the stages before moving onto the next – so, you can move on to changing your program to use classes before you have answered all of the questions for instance. Notice that there is a difference between showing a program works and showing a program runs.

c. **Your research**, no more than 500 words, then adding a section explaining how you got it to work in your program if applicable (*and the relevant sections of changed program code*).

d. **An evaluation** of what you have achieved over the semester.

Academic Misconduct.

There are a few who may try to cheat – not many. We take this very seriously and you may be asked to upload your work for plagiarism checks. If you do not upload you work when asked, you will be deemed to have not submitted your work and your score for the module will be zero. Penalties for cheating have ranged in the past from having the assessment set to zero to having the entire years marks set to zero. We don't like and we don't want cheats.

There is nothing wrong with talking about the examples and the exemplars, in fact it is good to work with others at a similar level **on the examples.** This is not the same as copying someones program code (see above). We don't catch you because you have the same right program. We catch you because you have the same wrong program! So there is help, and there is cheating. You need to be able to see the difference.

Understanding:

As you work through the assessment and when you are in the lectures, you need to make notes in your logbook – yes, you do need a logbook and, as in semester 1, there is an in-lab assessment in week 13 where you will only be able to bring in your logbook. I am sure that the experience of the first semester will convince you that this should be made up as you progress through the module and not be done at the last minute.

A logbook also becomes a resource, and you can look up various bits of information quickly when coding.

Use every resource you can to succeed, books, the internet, your pals, your tutors... but how do you measure success? Not by the final mark, but how useful your knowledge and skills will be when you start work, and how useful this learning is for future modules. People cheat to get the mark. They cheat themselves out of understanding.

Progress report:

IMPORTANT: Please fill in the table below and include it as the second page of your submission.

Description	Complete	Code Page No(s)	Evidence Page No(s)
I can open a file, read in the lines and split the lines into fields.			
I can get the program to calculate the answers to (some of) my questions			
I have modified the code to use Classes and Objects			
I have developed a GUI to interact with the user			
The GUI caches the values			
I have written my report on the research			
I have changed the code based on my research			
I have written an evaluation of what I did			
I have tested the classes in my program.			
I have printed and annotated evidence of the code running at each stage.			

√ Tick the bits completed
~ for partially completed
X for not present

Handing in work:

You work needs to be presented to your tutor in 2 forms. A soft copy so that tutors can run you program code if they so wish (a zipped up copy of all your programs + your report and any other documents) and your formal submission which should be printed out, presented in a folder or a single plastic envelope and submitted to L105, you will need to print a front sheet before submitting .... Please do not use multiple plastic covers or ring binders.

If you are using MSWord or LibreOffice Writer, please use a monospaced font such as `Courier` or `FreeMono` for all your Python source code and reduce the font size to remove any word wrap. All program code needs explanation (annotation) this can be in the form of comments or be handwritten onto the printouts.

All marking is done on the basis of evidence which means that even if you did the work, if it is not submitted, then you did not do it as far as we are concerned. Any work handed in late is automatically capped at 40% (28 marks for the coursework) unless you have special arrangements agreed or you submit a late submission form with evidence and it is accepted (forms from room L105). Any work handed in more than 2 weeks after the hand-in date will not be marked.

Software Development for Business 2

The Assessment Feedback and Mark Sheet (2013/14)

Name ________________________ Student No _________ Tutor______

	Description	Score	Max
1	You have opened the files using Python and read in the contents a line at a time.		2
2	You are able to show that the program could split your data files into separate fields and deal with them in an appropriate manner.		2
3	When you run your program it correctly works out the answers to your questions for all 3 files. (*2 marks each*)		14
4	...and there is clear annotated evidence of your program running in each of the three stages for all three files. *This means you have to somehow change the look of the output between stages 1 and 2!*		6
5	Your program only needs to read through the file once to get all of the answers		2
6	The user can select the file by entering the file name.		2
7	You structured your first version using functions.		2
8	You were able to able to structure your program using classes.		10
9	You have tested your classes using unit testing. (*Or appropriate demo classes*)		3
10	You have successfully used a GUI (tkinter) and it works, allowing the user to select the file, and by clicking a button, the answers are collected and displayed (*note: it is not enough to supply the answers that were collected previously. The program has to do the calculations.*)		9
11	The GUI caches the values so that subsequent reads are not needed.		2
12	You have written a simple report outlining what you found in your research.		5
13	You have applied what you have learnt in your research to **your** program and you have evidence to demonstrate this.		6
14	An evaluation of what you have learnt, the programs that you have developed and a reflection of your experiences programming during the semester.		5
15	Mark awarded for the in-class multiple choice.		10
16	Mark awarded for demonstrating understanding of supplied program code in the in-lab section of the coursework.		20
	Remember to include the progress report...		
	Total		100

Comments: