Review of the lectures 21 to 36

The second midterm exam will be closed book. You have to bring your own paper with you. Good examples of questions are quizzes and homework assigned at the end of each lecture.
The material is focused on lectures 21 to 35, although by now you should have a good working knowledge of writing Python programs. These 15 lectures cover 5 weeks, roughly divided in 5: (1) modules; (2) OOP (3) testing, exceptions, and complexity; (4) GUIs; and (5) applications.

This sheet contains some preliminary examples of questions which may help you prepare for the second midterm exam.

1. Design the modular structure of an online system to help planning a trip using public transport, for example with the CTA. What is at the bottom of the program? Describe the modules and their relations. Refer to the three key principles of a good design to explain why you choose this design. Start by thinking about only one subway or bus line.

2. One of the modular design principles we discussed is *low coupling and high cohesion*. Explain what this means and give examples indicating which properties hold. Why is this a desirable property of the design of large programs?

3. Compare the waterfall model to the spiral model of software development. Enumerate at least two relative advantages and disadvantages of each model. Which type of model would fit what type of company organization best?

4. Consider the problem of removing all occurrences of one word from a text file.

   (a) Describe the design of a program to remove a user given word for a text file whose name is given by the user as well. View the file as a sequence of characters. What data structures will you use? Show the control flow of the program using a flowchart.

   (b) Give the Python code for a program to remove all occurrences of a user given word.

5. The Unix command `cal` produces a calendar as (`$ is the command prompt`)

```
$ cal 4 2016
     April 2016
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

   Assume that this output of `cal 4 2016` is an input file for a Python program that should then write `April 2016 has 30 days`. Note that the last line is blank.

   (a) Draw the flowchart for the algorithm.

   (b) Give the Python program to implement the algorithm.

6. Explain the differences between software in the public domain and software available under GNU GPL. Give examples of public domain and GNU GPL software.

7. Suppose we want to administer a small online school. All administrative functions are taken care of by a program.

   (a) Use UML, defining class and use case diagrams to model the administrative programs. Students enroll in courses taught by faculty. Faculty administer their courses: create, delete and enter grades.

   (b) Give Python code to define the classes, i.e.: list the data and functional attributes.

8. A library contains music CDs and books. Describe a class `Item` that defines data attributes common to both music CDs and books. For the classes `CD` and `Book` inheriting from `Item`, define additional data attributes for both of them.

9. Give an example of rule #4 of a bug: "The software doesn't something that the specification doesn't mention, but should." Explain why the rule applies to your example.

10. The command `uname` returns information about the OS on Unix, Linux, and MacOS X systems. Write a function that returns the result of `uname`, and provide a handler in case we exceptionally use another OS.

11. Explain the differences between static blackbox and dynamic whitebox testing.

12. Consider the inner product of two lists of numbers, `A` and `B`, of length `n`, defined as `A[0]*B[0] + A[1]*B[1] + ... + A[n-1]*B[n-1]`. Count the number of arithmetical operations. Show that it is $O(n)$.

13. In what sense are the widgets Radiobutton, Checkbutton, and Scale similar? What are their differences? Give three examples of their most appropriate use.

14. Consider a system where entry is determined by a four digit access code (such as 9812).

    (a) Design a GUI where the user can *only* enter a four digit access code. By *only* we mean that (1) only 4 digits can be entered; and (2) only digits (i.e.: numbers in the range 0,1,..,9) are permitted. What kind of widgets will you use?

    (b) Write the constructor function in the object oriented implementation of this GUI.

    (c) Give the functional attributes in the class that implements this GUI.

15. Consider an ordering system in a fastfood restaurant selling burgers, hotdogs, and pizza slices. French fries are sold in small, medium, and large quantities.

    (a) Design a GUI to make ordering easy. The user should first see the total price of the order, before confirming the order. Define the layout and the actions of the GUI.

    (b) Give the object oriented design of the GUI, defining the data and functional attributes of the class. Write proper documentation strings.

    (c) Give Python code to implement this ordering system.

Note the policy on skipping exams: If an exam is missed, then greater weight will be placed on the final exam, especially on the material covered on the missing exam. Please be prepared when you show up for the exam. Skipping this midterm exam will make an application for an incomplete at the end of the semester very difficult.