

Team Project: 48024 Object Oriented Design Autumn 2007

The team project involves the design and construction of a small software system using a set of object oriented design techniques and the Java programming language. You are to conduct your project according to the "Miniature Unified Process" explained in the first class. As part of this process, you are expected to work in an iterative manner. That is, you should be evolving both your design and your implementation in stages, where each stage involves a design-code-test cycle. As you progress through the stages, the emphasis shifts from design in the earlier stages to coding and testing in the later stages.

Your team will be able to choose a project topic from a set of topics (see below).

The Team Project is worth 60 marks, and is composed of the following key assessable components:

- Project/design documentation, 30 marks
- Code implementation, 20 marks
- Teamwork, 5 marks
- Individual design thinking and contribution, 5 marks

The assessment criteria for the project will be accessible through *ReView*, an online criteria-based assessment tool. Make sure you take careful note of these assessment criteria whilst working on your project. Your project deliverables will be assessed against them.

Project/Design documentation (30 marks)

The project/design documentation is the written record of your analysis and design artefacts, produced through application of particular object-oriented analysis and design techniques. It is to be delivered incrementally as an article (s) on the Wiki.

Your article should contain the following:

Analysis: You will create and apply usage scenarios, use cases, and related artefacts to determine the scope and functionality of your program. All of your analysis artefacts must be reasonable, realistic, and consistent both with each other and with your "downstream" artefacts i.e. the UML models and the code. Together, they must paint a complete "picture" of what the software system is, what it does, and why.

Design of user interface: You will explore the design of the user interface through drawing design sketches of possible layouts and graphical interface components (such as buttons, sliders, windows, text, etc.). You may like to link these to specific scenarios. (**Note:** This part can be hand-drawn and stored in your logbook, unless you have easy access to a document scanner.)

UML models: You will choose and create appropriate UML models and contracts to describe and document the design of your program. The essential models are class and sequence diagrams. You will not be assessed on the quantity of your diagrams, but on the care, depth, and insight that you apply to choosing and designing your models. The models and contracts must be consistent with your analysis artefacts and with the program implementation. All diagrams must be accompanied by careful explanation -- a diagram without explanation is worthless.

Design of test cases: You will carefully design unit tests and system tests based on your contracts and use cases. Choose test cases that provide the most benefit in terms of locating potential weaknesses in the program. Carefully-chosen test cases will yield higher marks than a scatter-shot or exhaustive approach.

Quality and structure: You will be assessed on how well you have explained your design work. Careful structure, explanation, good writing, and thoughtful use of Wiki features will be considered in this part of the mark.

Citing other people's work: During this assignment, you will be looking up and referring to existing work in textbooks, online references, etc. You should have a *Citation* section in your Wiki article that lists any major sources of inspiration or reference, particularly regarding Java code. You should briefly describe how you used the source. Use the Harvard Referencing system (see the BELL website for guidelines).

Initial Design Milestone - Project Presentation and Feedback Session

This milestone represents the end of the Elaboration phase of the MUP. You should have an analysis of the project and initial design ready including usage scenarios, use cases, class diagram, class descriptions, and sequence diagrams. These artefacts do not need to be complete and definitive at this stage. But you should have made a reasonable start with all of them. A presentation and feedback session will be held mid-way through the semester to assess your initial design work and to provide constructive feedback on the design so far.

Clear, succinct verbal explanations of key aspects of your design are expected, with a walk-through of your project on the Wiki. Each team member must discuss some aspect of the design they worked on and will be individually assessed on their contribution so far. The instructor will ask questions, query design issues and give on-the-spot feedback

A peer review will be conducted by your class mates of presentation style, focussing on audience engagement, use of visual aids and diagrams, and clear delivery.

Any student not present or unable to give a satisfactory account of their involvement will be liable to incur a 10% deduction from their overall project documentation mark.

See the Subject Schedule for the date/time.

Documentation length and quality

Your written documentation must be clear and concise. Do not try and "blow up" your documentation with verbose explanations or (worse) text copied from other sources. Make every paragraph count. Marks will be deducted for excess verbosity or lack of clarity.

Similarly, don't produce dozens of diagrams without a good reason for each. Demonstrate your engineering skills by choosing your models and diagrams carefully, and explaining them carefully.

There is no hard and fast rule about length. If, however, your total documentation exceeds about 7,000 words, you should ask yourself whether your documentation is properly focused. You can also ask an instructor for feedback.

Code implementation (20 marks)

You will need to implement your design in Java to produce a working system that satisfies the functional requirements you have captured in your use cases and scenarios.

Each team will need to decide how to manage the division of programming work and integration of separate sections of code. You should decide this early on and document it on your Wiki article. Each student should include their name and student id in the header of any Java class they contributed to, along with a short description of their contribution to the class. Otherwise it will be assumed that you have made zero contribution to the code, and you will be awarded zero marks for this component accordingly.

You are encouraged to start coding early, at least to experiment with how you might implement some of your design ideas, and to familiarise yourselves with the Java language.

You are expected to follow standard style conventions for Java programming, especially with regards to naming, indentation, and commenting. See the style guide, <http://www.javaranch.com/style.jsp>

The project implementation is due in the second last week of semester. You must deliver a working program on a CD to the instructor at the beginning of the lab demonstration. If your program fails to compile or run when the instructor is assessing it, you will receive a reduced mark.

Lab Demonstration

Each team is expected to demonstrate their working program in the lab. You will be assessed on how well your program satisfies your functional requirements for the system.

Each student will demonstrate one system test case. You will need to refer to both the system test case and the associated use case during the demo. Make sure these items are clearly accessible on your Wiki.

You will also be assessed on the overall quality of your user interface. In particular, you will be assessed on the following points:

- Have you shown thought in the design and layout of your interface?
- Have you made effective use of Java UI support libraries?
- Do your system tests / use cases run smoothly and easily (from the point of view of the user)?

Teamwork (5 marks)

Team members are expected to collaborate on all aspects of the project. You are learning how to create and apply UML models and how to tie those in with your implementation and testing activities. You are also learning how to express your designs and communicate them to others. Thus, it is counter-productive to the subject goals for team members to focus only on one or two aspects of the project and to ignore others.

All team members should make sure that they actively participate in all project activities, and that they demonstrate this participation in their individual Logbook, the Wiki documentation, the source code and the project presentation and demonstration sessions.

Teamwork will also be assessed on your team's preparation, planning and conduct as evidenced through your use of recommended sheets from the "BELL Groupwork Workbook" (online ref: <http://www.bell.uts.edu.au/groupwork/resources>, Sheets 2, 3, 4, 8, 19).

Individual Design Thinking and Contribution (5 marks)

A fundamental objective of this subject is to develop your design thinking and ability to articulate your design ideas and decisions through a variety of visual and verbal techniques. Your logbook is used to assess your approach to the development of your project, and should include your design notes and sketches, meeting notes and actions, implementation notes and issues, test regime, record of time spent on specific tasks, and so on. The main purpose of keeping a logbook is to develop professional habits and skills in keeping technical notes, records of decisions and activities undertaken at the individual and team level. It will be checked regularly during tutorials and lab sessions. It is recommended that you use an exercise book and handwrite your entries. You may paste in copies of selected extracts from your Wiki documentation, as required.

The presentation and lab demonstration sessions will also be used to assess your ability to articulate your design thinking.

Sample marking scenarios

Scenario 1: Team member A works hard on only the implementation, while team member B works only on the documentation of UML models. The instructor expresses concern about the lack of integration and coherence between the UML models, the code, and the test effort. The instructor also points out that neither student has achieved what the subject is designed to do. The instructor awards the team a reduced mark for the project documentation and code implementation because of this lack of coherence, and awards a low mark for the Teamwork component because of their lack of participation in all aspects of the project.

Scenario 2: While team member A does a significant portion of the implementation, she realises that her team members will be left without any understanding of the connection between the UML and the Java program if she does it all. So, she works with her team members to encourage them to implement significant portions of the UML models, and supervises the integration and testing effort. In the process, she realizes that her understanding of the UML also had significant gaps. The team has also embraced the use of the BELL Groupwork resources and conducted their teamwork successfully through regular meetings, clear identification of roles, informative records of meetings, actions and decisions, and a fair allocation of tasks. They have also reflected thoughtfully on how their team performed, and how they might do better next time. The instructor is impressed by the quality and coherence of the design and implementation, and the whole team receives a high mark for the project documentation, code implementation and teamwork components.

Project topics

Topic A. Energy Monitor

The high levels of carbon dioxide emissions in the atmosphere are thought to have a major impact on global warming. Many countries around the world, including Australia, are now tackling the issue of climate change and global warming, through various campaigns aimed at reducing energy consumption and finding alternative “green” energy generation technologies such as wind and solar.

Part of the campaign around raising awareness and changing practices to reduce energy consumption includes the **Lights Off for Earth Hour** on 31st March, 7.30-8.30pm.

- <http://earthhour.smh.com.au/what-to-do-everyday.html>
- <http://www.cityofsydney.nsw.gov.au/3cbds>

Another initiative is to provide people with a portable handheld device that can monitor the energy usage in their home. You can plug in any appliance to the device and it will measure the power consumption and energy usage of that appliance. The device contains a software program that controls the monitoring of energy usage. The graphical user interface provides a set of functions for viewing chronological and real-time data. Power consumption is counted in kilowatt-hours. It can calculate electricity expenses by the day, week, month, or year. It displays volts, amps, and wattage within 0.2 percent accuracy. You can check the quality of your power by monitoring voltage, line frequency, and power factor. For each appliance you can set up a unique record to store measurements over time. This device is intended for an Australian market. We have a 240V power supply.

It is your task to design and build a prototype for this software program. It is up to you how you design the graphical user interface and what features you provide. You will need to provide simulated input data for each appliance. In the future, this device may be extended to include monitoring of power consumption directly from the mains power supply, so that lighting and other equipment that is hardwired into the mains power board can be included.

You may like to explore the following links to see what features these existing devices offer.

- <http://www.thinkgeek.com/gadgets/electronic/7657>
- http://www.treehugger.com/files/2006/07/wattson_monitor.php
- <http://www.theenergydetective.com/features.asp>

Topic B. Head-controlled Music Player

Some severely physically disabled people lack the use of their limbs, but still have control of their head movements. A need has been identified for a device that can allow a person with only control of their head to interact with a sound system for playing recorded music.

The vision is for this device to consist of a small, wearable accelerometer that measures the direction, tilt and acceleration of the person's head in order to derive input control commands. The wearable device communicates over wireless with the sound system to control the selection, playback, volume, and so forth, of music. The sound system contains a graphical user interface that displays the current track, a playlist, duration of track, etc. The system can play tracks in .wav and .mp3 format.

The graphical user interface also contains a separate panel that shows a visual indication of the user's head movements, and associated commands. You need to explore what kinds of head movements make sense for this application, and what commands they will represent.

For this project you will develop a software prototype for this system. You will need to use the Java Sound API. It is up to you how you design the graphical user interface and what features you provide. It is envisaged that future applications may want to reuse the head movement monitoring part of the program, and provide different mappings from head movements to commands. You will need to simulate the accelerometer inputs (unless you happen to have an accelerometer! Although interfacing to the physical device is beyond the scope of this subject.) – you can do this by providing a mouse-operated GUI that mimics the head movements with some combination of mouse actions.

You may like to read some publications of work done in the related area of wheelchair control.

- P.B. Taylor, H.T. Nguyen, "Performance of a head-movement interface for wheelchair control", Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE, Vol.2, 17-21 Sept. 2003, p. 1590-1593.
- H.V. Christensen, J.C. Garcia, "Infrared Non-Contact Head Sensor, for Control of Wheelchair Movements", IOS Press, 2003 (online: vbn.aau.dk/fbspretrieve/529548/aaate_article.pdf, accessed 1st March 2007)

Here is a description of how an accelerometer works:

- <http://www.dimensionengineering.com/accelerometers.htm>