

Assignments 4 and 5: Object-Oriented Software Implementation (Group Work)

Robert S. Laramee

March 14, 2012

Contents

1 Problem Statement and Overview	1
2 Assignment 4 Partial Implementation Tasks	2
2.1 Assignment 4 Deadlines: Electronic and Printed	2
2.2 Partial Object-Oriented Implementation for Assignment 4	2
2.3 Assignment 4 Submission: Partial Implementation and Group Report	2
2.3.1 Minutes Protocol (5%)	2
2.3.2 Partial Application Implementation ByteCode (50%)	3
2.3.3 Partial Application Implementation Source Code (20%)	3
2.3.4 Application Implementation V1.0 Documentation (10%)	3
2.3.5 Demo via Screen Capture (10%)	3
2.3.6 Group Report (5%)	3
2.3.7 Digital Copies of Files	3
3 Assignment 5, Complete Implementation Tasks	4
3.1 Assignment 5 Deadlines: Electronic and Printed	4
3.2 Assessment Report for Assignment 5	4
3.3 Assignment 5 Implementation	5
3.4 Assignment 5 Submission: Assessment Report, Complete Implementation, and Group Report	5
3.4.1 Group Minutes (5%)	5
3.4.2 Software Assessment Report (10%)	5
3.4.3 Complete Application Implementation ByteCode (50%)	5
3.4.4 Complete Application Implementation Source Code (10%)	5
3.4.5 Application Implementation Documentation (10%)	5
3.4.6 Demo via Screen Capture (10%)	5
3.4.7 Group Report (5%):	6
3.4.8 Digital Copies of Files	6
4 Assignment 6 Preview	6
4.1 Assignment 6 Deadlines: Electronic and Printed	6
5 Object-Oriented Implementation for Assignments 4, 5 and 6	6
6 Implementation Documentation using Doxygen for Assignments 4, 5, and 6	7
7 Group Report for Assignments 4 and 5	7
8 Project Hints	7
9 Learning Outcomes and Transferable Skills	8

1 Problem Statement and Overview

1 In this assignment, each group will build a partial application called a digital organizer, a sample image of which is shown in the requirements specification document as part of the previous assignment. A detailed list of the requirements specifications is given in the Functional Specification Section of the previous assignment.

Recall, the software design and implementation are broken down into multiple phases:

1. **Assignment 3: Design (Group Work):** Each group proposed an object-oriented design for the *entire* application. The design requirements are described in detail in the previous assignment.
2. **Assignment 4: Partial Implementation (Group Work):** Each group implements some of the features described in the feature specification given in the previous assignment.
3. **Assignment 5: Hand-Off and Assessment (Group Work):** Each group then hands their full design and partial implementation to another group. Correspondingly, each group takes over another groups' design and partial implementation. Each group then writes a report assessing the project work they have taken over with respect to the quality of the design, implementation, and testing. A detailed description of this phase will be provided.
4. **Assignment 5: Complete Implementation (Group Work):** Each group implements V2.0 of the system, i.e., the complete set of features described in the functional specification.
5. **Assignment 6: Re-Engineering Implementation (Individual Work):** Assignment 6 is a re-

engineering task where each individual implements features which are not yet defined. The full description of this assignment will be provided in a future document.

You and your team are now in the partial implementation phase.

2 Assignment 4 Partial Implementation Tasks

Must-Have Features: Version 1.0 for this assignment is a partial implementation. It must include at least:

1. a monthly view
2. three event types (1), (2), (3)
3. The ability to (1) add and (3) delete contacts to/from the address book.

The event and contacts should be persistent. In other words it should be possible to save them to disk and re-load them.

2.1 Assignment 4 Deadlines: Electronic and Printed

Number of Credits: 35% of a 15 credit module

Recommended Hours: Approximately 50 hours (per group member)

Important Deadlines :

1. **13 February** by 11:00am: copy of first minutes of meeting
2. **Partial Implementation (Section 2.2) and Group Report (Section 3.4.7)**
 - (a) **27 Feb** at 11:00am: online submission of partial implementation software, demo(s), doxygen output, and group report and paper/printed submission of group report

Paper/printed submissions can be dropped into the coursework submission drop-box or given to the Departmental Student Secretary in room 206 of Faraday Building.

2.2 Partial Object-Oriented Implementation for Assignment 4

2.3 Assignment 4 Submission: Partial Implementation and Group Report

This section describes which files need to be submitted for the assignment and how they should be submitted. Remember that each member of the team is required to implement two (or more) classes. Also, each team member must be able to demonstrate that their classes are working properly to the other team members through the use of unit tests.

This assignment is group work. Electronic copies of each report document, Java file, and JAR (Java Archive) file are kept online using the computer science web server. See the module web page for instructions on how to place files on the computer science web server.

One member of each group, the **Planning and Quality Manager**, is required to create a folder called `cs235groupNa4`, where *N* is replaced with their group number, in their `public_html` folder. The `cs235groupNa4` online folder contains (and organizes) digital copies of all of the files that compose this assignment. In this way, anyone with a link to this folder can read and download all of the files that make up this assignment.

2.3.1 Minutes Protocol (5%)

A copy of your group minutes for three (or more) meetings held before the assignment deadline. Your minutes files should be called "*GroupNminutesD-monYr.txt*" where *N* is replaced by your group number, *D* is replaced by the calendar day your group met, *mon* indicates the month that the group met, and *Yr* indicates the year, e.g., *Group1minutes23oct09.txt*. (You will be assessed on this.) Send a *link* to your first minutes to customer Bob as an email to reassure him that you are working hard on his product. A minimum of three minutes of meeting protocol is required and are submitted on behalf of the group by one of its members. *The first minutes has a special interim deadline before the other files.* See the given interim deadline. In addition, a digital copy of each minutes file is placed in a folder called `minutes` that resides in the `cs235groupNa4` folder described above.

2.3.2 Partial Application Implementation Byte-Code (50%)

The application itself must also be submitted online. The main executable file should be called “*GroupNdigitalOrganizer.class*” where *N* is replaced by your group number. Pack your class files together in a Java Archive (.jar) file. Digital copies of implementation source files are placed on the CS web server in a folder called `implementation` that resides in the `cs235groupNa4` folder described above. Do not place a copy of every individual class file online, just the main executable and the Java archive file.

2.3.3 Partial Application Implementation Source Code (20%)

The application source code must also be submitted. The source file should be called “*GroupNsourceCode.zip*” where *N* is replaced by your group number. We recommend zipping the Java source code files together and storing them as one file online. You can also use tar to pack your source files together, e.g., you can log onto the CS server and type:

```
%> tar -cvf GroupNsourceCode.tar \
  implementation/*
```

Do not create a .rar file as these are not platform independent. Digital copies of all Java source files are placed on the CS web server in a folder called `implementation` that resides in the `cs235groupNa4` folder described above. The implementation follows the rules given in Section 5.

2.3.4 Application Implementation V1.0 Documentation (10%)

A copy of your application implementation documentation submitted online is required by the deadline(s) indicated above. Doxygen produces a series of linked HTML web pages in order to facilitate the browsing of project implementations. Refer to Section 6 for more on this topic. The HTML web pages produced by doxygen are placed on the CS web server in a folder called `doxygen` that resides in the `cs235groupNa4` folder described above.

2.3.5 Demo via Screen Capture (10%)

Each group uses screen capturing software to demonstrate the features of their application. Several links to (free) screen capturing software are given on the module web page.

The file(s) is named after the feature(s) being demonstrated e.g., `group5digitalOrganizerDemo.mpg`. The movie files are saved in MPEG format. You may use as many screen capture files as necessary to capture the features of your application. One movie that captures all the features is ideal.

Your animated screen captures are also placed on the CS web server in a folder called `demo` that resides in the `cs235groupNa4` folder described above.

2.3.6 Group Report (5%)

A description of what and how each group member contributed to the project is submitted. More detail about the group report content is given in Section 7. The file is called “*GroupNmemberContributions.pdf*” where *N* is replaced by your group number. A digital copy of the group report is placed on the CS web server in a folder called `groupReport` that resides in the `cs235groupNa4` folder described above.

The group report also tells the reader: (1) where they can download the project files from (the URL) and (2) how to compile the code to produce the data visualizer application. *A printed copy of the group report is also submitted.* Only one cover sheet is required on behalf of the group.

2.3.7 Digital Copies of Files

Thus, when completing the submission of the assignment, the Planning and Quality Manager has a directory structure in their `public_html` folder that looks like this:

```
.../public_html/cs235groupNa4/
  demo/
  design/
  doxygen/
  groupReport/
  implementation/
  minutes/
```

Make sure that all of the files and folders are accessible (readable and executable) to anyone who has a link to your teams’ `public_html` folder: e.g.,

```
%> chmod -R ugo+rx *
```

Place a PDF copy of your groups’ design document in the design folder. Points will be deducted for those

submissions that do not follow the file naming conventions and required file formats.

Also, customer Bob may ask for a demonstration of the software by a group if he needs extra reassurance that the program is working as advertised.

If you are worried about someone seeing your submission before the deadline, you may use the following command to prevent others from reading it: e.g.,

```
%> chmod -R u=rwx *
```

Just make sure to make all files readable by everyone on the assignment deadline.

3 Assignment 5, Complete Implementation Tasks

This assignment starts off with an evaluation phase of the previous assignment. The assessment report is described in detail in Section 3.2.

3.1 Assignment 5 Deadlines: Electronic and Printed

Number of Credits: 35% of a 15 credit module

Recommended Hours: Approximately 50 hours (per group member)

Important Deadlines :

1. **2 March** by 11:00am : copy of minutes of meeting

2. **Assessment Report (Section 3.2)**

5 March by 11:00am: paper/printed and electronic (online) submissions of assessment of assignment 4 group work

3. **Implementation (Section 2.2) and Group Report (Section 3.4.7)**

~~26 March~~ **2 April** by 11:00am: online submission of complete implementation software, doxygen output, demo(s) and group report and paper/printed submission of complete implementation group report

Paper/printed submissions can be dropped into the coursework submission drop-box or given to the Departmental Student Secretary in room 206 of Faraday Building.

3.2 Assessment Report for Assignment 5

During the week after the submission of A4, each group assesses the software design and implementation they have received and provides a report of their assessment. The report, no more than 10 pages, answers questions such as the following:

1. Have all the required files and documents been provided, e.g., design document, group report, minutes, source code, class files, and web pages produced by doxygen?
2. Have all the required files and documents been provided on time? If not, how late was the group in delivering the product?
3. Is the software design complete? Does it make sense? Have decisions been justified? Have at least three class hierarchies been identified and described? Have at least two-three sub-systems been identified and described?
4. Does the source code compile and provide a running application?
5. How closely does the source code conform to the coding conventions?
6. Do the functions specified in the group report actually work as advertised?
7. How well does the design match up with the implementation?
8. How robust is the software? What tests does it pass and when does it fail?
9. How does the software rate in terms of usability? Is it intuitive?
10. How complete is the doxygen output? Are both the short and detailed class descriptions there? Are both the class hierarchy and collaboration diagrams there? Is the source code there?

These questions are here only to give each team the ideas behind the assessment report (and to get you thinking about evaluation). More detail about the assessment will be given on the module web page. Also, a sample copy of the assessment report will be given before the partial implementation deadline such that each group knows how their application will be tested and evaluated. The team delivering a late product will

lose points according to how many days after the deadline the requirements are fulfilled. Those lost points will then be awarded to the recipient team.

3.3 Assignment 5 Implementation

Assignment 5 implements the remaining features not specified above in Section 2.2. In other words, this assignment implements everything described in the design requirements which documents the complete functional specification.

3.4 Assignment 5 Submission: Assessment Report, Complete Implementation, and Group Report

Again, electronic copies of each report file, Java file, and JAR file are kept online using the computer science web server similar to the previous assignment.

The **Planning and Quality Manager** of each group creates a folder called `cs235groupNa5`, where N is replaced with their group number, in their `public.html` folder. The `cs235groupNa5` online folder contains (and organizes) digital copies of all of the files that compose this assignment.

3.4.1 Group Minutes (5%)

A copy of your group minutes for three (or more) meetings held before the assignment deadline is required. The protocol from the previous assignments is used with the same file naming and format conventions. A minimum of three minutes of meeting protocol are required and are submitted on behalf of the group by one of its members. The first minutes has a special interim deadline before the other files. See the given interim deadline. In addition, a digital copy of each minutes file is placed in a folder called `minutes` that resides in the `cs235groupNa5` folder described above.

3.4.2 Software Assessment Report (10%)

A copy of your Software Assessment Report online *and in printed form* is required by the deadlines indicated above. More detail on this report is given in Section 3.2. The file should be called “*GroupNassessmen-tReport.pdf*” where N is replaced by your group number. PDF format is strongly encouraged. In addition, a digital copy of this report is placed on the CS web server in a folder called `assessment` that resides in the `cs235groupNa5` folder described above.

3.4.3 Complete Application Implementation ByteCode (50%)

The next version of the application itself must also be submitted online. The main executable file should be called “*GroupNdigitalOrganizerV2.class*” where N is replaced by your group number. Pack your class files together into a Java Archive (.jar) file. Digital copy the class files are packed together as a JAR file and placed on the CS web server in a folder called `implementation` that resides in the `cs235groupNa5` folder described above.

3.4.4 Complete Application Implementation Source Code (10%)

The application source code must also be submitted. The source file should be called “*GroupNsource-CodeV2.zip*” where N is replaced by your group number. We recommend zipping the Java source code files together and storing them as one file online. You can also use tar to pack your source files together, e.g., you can log onto the CS server and type:

```
%> tar -cvf GroupNsourceCode.tar \
  implementation/*
```

Do not create a .rar file as these are not platform independent. Digital copies of all Java source files are placed on the CS web server in a folder called `implementation` that resides in the `cs235groupNa5` folder described above. Remember, the implementation follows the rules given in Section 5.

3.4.5 Application Implementation Documentation (10%)

The doxygen documentation submitted online is required by the deadline(s) indicated. Refer to Section 6, for more on this topic. The HTML web pages produced by doxygen are placed on the CS web server in a folder called `doxygen` that resides in the `cs235groupNa5` folder described above.

3.4.6 Demo via Screen Capture (10%)

Similar to the previous assignment, each group uses screen capturing software to demonstrate the features of their application.

The file(s) is named after the feature(s) being demonstrated e.g., `dayWeekMonthViews.mpg`. The movie files are saved in MPEG format. You may use as many

screen capture files as necessary to capture the features of your application, however, one movie file is ideal.

Your animated screen capture(s) is also placed on the CS web server in a folder called `demo` that resides in the `cs235groupNa5` folder described above.

3.4.7 Group Report (5%):

A description of what and how each group member contributed to the project is submitted. More detail about the group report is given in Section 7. The file is called “*GroupNmemberContributions.pdf*” where *N* is replaced by your group number. A digital copy of the group report is placed on the CS web server in a folder called `groupReport` that resides in the `cs235groupNa5` folder described above.

The group report also tells the reader: (1) where they can download the project files from (the URL) and (2) how to compile the code to produce version 2.0 of the application. A printed copy of the group report is also submitted.

3.4.8 Digital Copies of Files

Again, when completing the submission of this assignment, the Planning and Quality Manager has a directory structure in their `public_html` folder that looks like this:

```
.../public_html/cs235groupNa5/
    assessment/
    demo/
    doxygen/
    groupReport/
    implementation/
    minutes/
```

Make sure that all of the files and folders are accessible (readable and executable) to anyone who has a link to your teams’ `public_html` folder: e.g.,

```
%> chmod -R ugo+rx *
```

Customer Bob may ask for a demonstration of the software by a group if he needs extra reassurance that the program is working as advertised.

4 Assignment 6 Preview

Assignment 6 is a re-engineering task. The feature specification given in Assignment 3 (Design) will

change. Each individual will then implement those changes. The better the teams are at designing and implementing assignments 4 and 5, the easier this task will be. Assignment 6 also encourages each group member to be highly involved in the group work in assignments 4 and 5. The more familiar you are with the previous work, the easier assignment 6 will be.

4.1 Assignment 6 Deadlines: Electronic and Printed

Number of Credits: 30% of a 15 credit module

Recommended Hours: Approximately 40 hours (individual work)

Important Deadlines :

1. **8 May** by 11:00 am: printed and online submission of A6 implementation and report

5 Object-Oriented Implementation for Assignments 4, 5 and 6

In parallel and after the design, the application also needs to be implemented in Java and Java Swing [1, 2]. Several helpful links to Java resources are given on the module web page.

The implementation needs to adhere to some rules. Customer Bob requires these rules because he wants a good product to be delivered. When he looks under the hood, he does not want to see a pile of scrap metal, but rather a finely tuned, carefully crafted machine.

1. **Coding Conventions:** Your team is required to follow *Bob’s Concise Coding Conventions* [5]. See the module web page for a copy of this document.
2. **Code Comments:** Your team required to use Doxygen to comment your source code. Your code must be commented according to the guidelines given in *Bob’s Concise Introduction to Doxygen* [4]. The doxygen program is free and available in the Linux lab of the computer science department. See the module web page for a copy of *Bob’s Concise Introduction to Doxygen* [4].

Teams often lose points because they do not follow the coding conventions. or they do not submit fully-completed doxygen documentation.

6 Implementation Documentation using Doxygen for Assignments 4, 5, and 6

Of course the actual implementation may change with respect to the original design. Thus, your team will also produce application implementation documentation as part of your project. The good news is that the application documentation is generated automatically using Doxygen. Use Doxygen to generate HTML web pages that provide:

1. A list of classes with a short description,
2. A list of classes with a detailed description including a list of all methods (both public and private) with the attributes defined in Bob's Concise Introduction to Doxygen [4],
3. Both class hierarchy diagrams and collaboration diagrams (generated automatically by Doxygen).
4. The source code of each class.

Points will be subtracted if any of these components is absent. Instructions on how to produce this output is provided by Bob's Concise Introduction to Doxygen [4] and `doxygen.org`.

7 Group Report for Assignments 4 and 5

Each group member is obliged to contribute two classes, both design and implementation, to assignments 4 and 5. They must be able to demonstrate that their classes work to the other group members using unit tests.

The group report indicates which group member(s) contributed to the project and how. The group report, no more than 5 pages, describes who contributed to:

1. classes and sub-systems in the implementation,
2. testing: both unit testing and integration testing.

Each manager writes part of the report. Thus, there are sections written by the (1) Customer Interface Manager, (2) Design Manager, (3) Implementation Manager, (4) Test Manager, and (5) Planning and Quality Manager. Each section answers the questions that

each manager is responsible for addressing. The questions are provided in the software engineering lecture on *Team Roles and Group Work* [3].

Every student needs programming practice. This is one of the main reasons we are here. The group report must contain a list of each group member along with the classes they implemented. Group members who have not implemented any classes will be penalized and may not receive any credit. Group members who are strong in implementation help those that are weaker by teaching them and giving them guidance on how to implement a class or write methods in Java.

The group report also lists which features were:

1. implemented and are working properly,
2. implemented and are not working properly,
3. not implemented with an explanation of why they were not implemented.

The report also informs the reader if any unexpected problems arose during the course of the assignment. Likewise, the group experienced success in some areas, both expected and unexpected, this is included.

The group report also tells the reader: (1) where they can download the project files from (the URL) and (2) how to compile the code to produce the application.

8 Project Hints

1. Customer Bob recommends storing the data as human-readable ASCII or Unicode and is CSV (comma-separated values) format.
2. Remember that it is best to have a robust and stable application with fewer features than an unstable, full-of-bugs program with many features.
3. Test your application thoroughly. This includes unit testing, sub-system testing, and integration testing. Your program should be able to handle random input without crashing. Customer Bob is going to test your program with some very strange tests, e.g., negative letters and many other strange things. If the program crashes, less money will be paid.

4. This is likely the most difficult and costly assignment you have undertaken so far in your degree (or life for that matter), so make sure to get started early.
5. All group members are expected to contribute to all phases of the development including design, implementation, and testing. Group members who are weak at programming may get help from other members who are better at it. The same is true for design and testing. The author fields in both the class design and implementation reflect individual group member contributions.
6. Allow your group extra time to learn how to use Doxygen.
7. More hints for the assignments will be given out in lectures.
8. Ask questions in your tutorial.
9. Why not use skype for some of your group meetings?
10. Do not use email to resolve conflict.
11. In order to get a distinction level score on assignments 4 and 5, you have to do very well or excellent for each component of the deliverables, e.g., (1) design, (2) implementation, (3) testing, (4) documentation. If any one of these aspects is weak, your project cannot receive a distinction level score.

9 Learning Outcomes and Transferable Skills

Learning Outcomes: Students will gain an understanding of the principles of software engineering; an understanding of the key HCI concepts in the context of system evaluation and design; the ability to design and evaluate GUIs; an understanding of object-oriented programming concepts, and knowledge of their applications in software design and engineering processes; the ability to build GUIs and skills of event-driven programming; experience and appreciation of group work; skills of project management.

Transferable Skills: Problem solving through analysis and abstract reasoning. The ability to read critically, to precis and judge information. Experience and

appreciation of team work, time management, project management, and risk assessment. Skills in written communication and documentation. The ability to learn and use computer systems and software packages effectively.

References

- [1] P.J. Deitel and H.M. Deitel. Java for Programmers. Prentice Hall, 2009.
- [2] T. Gaddis and G. Muganda. Starting Out with Java, From Control Structures through Data Structures. Addison Wesley, first edition, 2007.
- [3] W.S. Humphrey. TSP: Coaching Development Teams. Addison-Wesley, 2006.
- [4] R. S. Laramee. Bob's Concise Introduction to Doxygen. Technical report, The Visual and Interactive Computing Group, Computer Science Department, Swansea University, Wales, UK, 2007. (available online).
- [5] R.S. Laramee. Bob's Concise Coding Conventions (C^3). Advances in Computer Science and Engineering (ACSE), 4(1):23–26, 2010. (available online).