



SE2890: Software Engineering Practices

Lab 2: The Bank of MSOE

Due: 11:00 PM CDT March 20, 2012

1. Objectives

- Apply effort estimation techniques to estimate the effort necessary to complete a software development project
- Critique the accuracy of software engineering estimates at the completion of a software development project

2. Prelab

Before starting this lab, you will need to install the Enterprise Architect software, available from the course website. If you have never installed Enterprise Architect on your system before, simply install using the executable and the attached license. If you have previously installed EA, then you will need to uninstall the previous version. This may require the usage of the Microsoft uninstaller.

After installing the tool (or before), watch the videos on using EA to create a class diagram and a sequence diagram.

3. Overview

The purpose of this lab is to again practice the usage of an object oriented design tool, namely Enterprise Architect. You will also perform some rudimentary effort estimates based on the results of last week's lab.

4. Detailed Instructions

In this lab, you will construct a small application to model the behavior of a bank. Namely, your bank has the ability to create accounts, deposit into accounts, support balance inquiries, and withdraw from accounts. Your program will need to include three classes. Each class will have the following responsibilities:

4.1. *Bank Account class*

- Create accounts with an initial deposit and an initial account number
 - Hint: Initial deposit must be \$1.00 or more.
- Support deposits, withdrawals, and balance inquiries.
 - Do not allow over-drafts.
- Deposit values must be positive.
- Provide support for an integer account number.

4.2. *Bank class*

- Keeps track of all the accounts using an existing Java data structure.
 - (Any data structure can be used that you talked about in Data Structures. Certain ones will work better than others given the assignment.)
- Allows the creation of accounts with an initial deposit and an account number.
 - (Hint: Check to make certain a given account number does not already exist before creating a new one.)
- Support deposits, withdrawals, and balance inquiries for any account on record.



- (Hint: Check to make sure an account exists before trying to run a report on it.)
- Report on the total deposits within the bank.

4.3. User Interface (Main program)

The user interface shall handle all interactions with the user. Namely, this shall include

- Prompting the user for the name of the command file to load
- Parsing the command file
- Displaying the results on the console
 - NOTE: NO class outside of the user interface shall ever write to the screen nor query the user.
- Displaying any error messages
 - NOTE: No class outside of the user interface shall directly write to the console any error messages.

When the program starts up, the user shall be prompted to enter a bank command file. This file will list a set of commands that are to be processed. The format for the commands is given in Figure 1. Command parameters are separated by spaces or tabs. For the inquiry command, the presence of an account number indicates that the inquiry is in regards to the assets of an individual account while the lack of an account number indicates the inquiry is in regards to the total bank assets. A sample file is given in Figure 2, and the output of running this program on a computer is shown in Figure 3.

```
<C> <Account Number> <Initial Deposit>  
<D> <Account Number> <Deposit Amount>  
<W> <Account Number> <Withdraw Amount>  
<I> <Account Number>  
<I>
```

Figure 1 Transactions command format.

```
C 10 100.00  
C 72 50.00  
I  
C 10 100.00  
D 10 100.00  
I 10  
W 72 51.00  
W 72 49.99  
I 72  
I
```

Figure 2 Sample transaction file.



```
Enter the name of the file to process.  
bankTest.txt  
Bank account 10 successfully created.  
Bank account 72 successfully created.  
Bank of MSOE has $ 150.00 in total Assets  
Bank Account Creation Error: Duplicate account number. Account already exists.  
C 10 100.00  
Bank deposit to account 10 successfully completed.  
Account 10 has $ 200.00 in total Assets  
Bank Withdraw Failed: Insufficient funds for withdraw.  
W 72 51.00  
Bank withdraw from account 72 successfully completed.  
Account 72 has $ 0.01 in total Assets  
Bank of MSOE has $ 200.01 in total Assets
```

Figure 3 Sample output from running the program.

5. Lab Process

As with the previous lab, adherence to the process is as important as completing the project. **FAILURE TO FOLLOW THESE INSTRUCTIONS WHILE STILL DELIVERING A FINISHED PRODUCT WILL IMPACT YOUR GRADE.**

5.1. Planning

As with last week, the process will begin with planning, and your time tracking should start as soon as the planning phase commences.

Start the planning phase by looking at each of the classes you are going to create. For each class, estimate the number of methods that you will need, as well as the relative size of each of the methods (small, medium, large)¹. From this, estimate how many lines of code will be in each of the classes and record this in the spreadsheet.

Next, using the estimate for the number of lines of code that you are to create, determine how long it will take you to complete this project. This can be done by taking the LOC count and dividing it by your productivity in LOC/Hour. This will result in the number of hours that this project is estimated to take to complete. Using this number and the percentages from the previous lab, determine how long it will take to complete each phase of the project.

Using this information, develop a workplan for completing your project. How much time will you work each day between now and the due date of the assignment, and on what aspects will you be working during each of those days.

Once you have completed this, print your plan out to pdf format and upload it to the website as lab2Plan.pdf.

¹ Small methods are ½ the average size, medium methods are average size, and large methods are twice the average size.



5.2. Design Phase

Once you have completed planning, design your code using Enterprise Architect. Your design will most likely consist of 3 classes on a class diagram as well as a sequence diagram showing the calls between classes when processing transactions.

Before you start using EA, you might find it advantageous to sketch preliminary designs using a pencil and paper. This might help to get your thoughts together before you deal with the tool.

When you are done with this phase, save your design file.

Mark the completion time for the design phase, and record the time spent in this phase.

5.3. Implementation Phase

The implementation phase is the same as the implementation phase from last lab. Simply translate the design into source code.

Your code shall be implemented in the package <your id> where <your id> is the id you use to log into Blackboard, E-mail, etc.

As was done last week, if you missed a method, add it to the design, but make sure you either save to a different filename or use a different color to add it to your diagram. Keep a simple tally of how many changes you had to make to your design phase during the implementation phase.

Once the implementation of your code is completed, save the code, but **DO NOT COMPILE IT**.

5.4. Review Phase

Before compiling your code, read through your code again. Are there any mistakes you see? Are there any problems that have occurred when you translated your design into source code that you recognize by “proofreading” your code? If so, fix them here and keep a simple tally of the count of the errors fixed in this phase.

5.5. Compilation Phase

As was done in the previous lab, click on the project menu and build to build your code. If there is a compilation error, fix it in this phase, all while keeping a tally of the number of errors fixed in this phase.

5.6. Testing Phase

Once your code compiles, test it by running a set of tests to verify that your program is working properly. If there is a problem that requires a fix, fix it while keeping a tally of the number of errors fixed in this phase.



5.7. Post Mortem Phase

In this phase, you are to take a look at the performance on your project.

Start by measuring the size of the program you created and counting the number of methods. Then, answer the following questions:

1. How long did you spend in each of the phases (planning, design, implementation, review, compiling, testing, postmortem), and what percentage of total project time does this represent?
2. How many problems did you resolve in each of the phases?
3. What was your productivity (Lines of code / hour) for the entire project? (This is calculated by dividing the LOC count of your project by the total project time.)
4. How accurate was your estimate?
5. How close to your initial workplan was your actual effort?
6. How might you improve your performance?

6. Deliverables

Deliverables are due to be submitted by 11:00 PM CDT March 20, 2012. Deliverables shall be submitted electronically through the course website.

6.1. Source code

Submit your completed source code through the course website. This should only include the java files. No other files should be submitted. It is best to zip them into a single .zip file with the format <yourid>_lab2_code.zip.

6.2. Lab Report

The lab report should consist of the following:

1. Title page. This should contain your name, section, instructor's name, Date, and the assignment number / description.
2. Design. Submit copies of your design. This should include both your initial design (made in the design phase) as well as the final design that you used for implementation. Designs should be pasted into your report from EA as figures.
3. Time Logs. Paste a copy of the Excel spreadsheet you completed with your time logs, showing start and stop times as well as interruption times.
4. Defect Information – Submit in tabular format a summary of the mistakes you found by phase.
5. A brief description of any problems you encountered or questions you have regarding the lab.
6. Questions. Answer the questions posed in the post-mortem phase.
7. Conclusions. State what you learned from this experience.

If you have any questions, consult your instructor.