

CS389 Software Engineering Weekly Assignments

Spring 2005

Week 1: Jan. 23 – Jan 29

1. Read Chapter 1 of the textbook.
2. Review, understand, and run Java source code files ShowException.java, ReadInputDemo.java, StringTokenizerDemo.java, and CollectionDemo.java.
3. Answer the following question by responding to my Weekly Question 1 thread
 - Can we ever build large software applications with the same level of confidence about their quality as civil engineers can on their bridges? Why?

Week 2: Jan. 30 – Feb. 5

1. Read Chapter 2 of the textbook.
2. Review, understand, and run Java code files InheritanceDemo.java, InterfaceDemo.java, StaticDemo.java, TestThreads.java, PiggyBankWithoutSynch.java, and PiggyBankWithSynch.
3. Write a Java program to determine whether the input string is a palindrome. A palindrome is a string that reads the same no matter you read it from left to right or from right to left. You should take advantage of class java.util.Stack. You may need to use some methods of String or Stack that are new to you. Learn to use Java online documentation at <http://java.sun.com/j2se/1.4.2/docs/api/> to find the necessary information. (Hint: You need String's methods length() and charAt(int). You need to convert char into Character object, with Character's constructor, and convert a Character object into a char value by Character's method charValue(). You need to know how to cast a general java.lang.Object object into a Character object.)
4. Answer the following question to my Weekly Question 1 thread
 - Can we ever prove that a program is correct by conducting thorough debugging and testing? If yes, why? If not, why all companies spend lots of resources in debugging and testing?

Week 3: Feb. 6 – Feb. 12

1. Read Chapter 3 of the textbook.
2. Read, understand, and run Java programs "StaticDemo.java", "ReflectionDemo.java", "AreaServer.java", and "AreaClient.java".
3. Write client/server programs "AccuServer.java" and "AccuClient.java" where the server supports the function of a method "int addTo(int v)" for reporting the accumulated argument values for the method since the start of the client session. The server should be able to serve up to 100 concurrent clients a time. The programs should use a default port number for the server's server socket, and also

allow the users to provide a different port number on the command-line at server/client program startup.

4. Discuss on the question: "Who should own the software source code?" Consider the question in the context of protecting intellectual property, encouraging market competition, and software quality control. The subquestions include: (a) which stakeholders of a software process should own the software source code; (b) what kind of source code ownership could promote fair business competitions; (c) if we buy binary code (like Java jar files) without its source code, how can we have confidence in its quality?
5. Set up two groups. each 5-6 student will make a group, and each group should select a group leader. Each group should have balanced skills in program design and implementation, leadership and project management, and technical writing.
6. As part of week 4's assignment, group 1 should prepare to present OCSF files "AbstractServer.java" and "ConnectionToClient.java" on Feb. 17, and group 2 should prepare to present OCSF files "AbstractClient.java", "ChatClient.java", "ChatIF.java", "ClientConsole.java", and "EchoServer.java" on Feb. 17.

Week 4: Feb. 13 – Feb. 19

1. Complete the reading of Chapter 3 of the textbook, listen and watch the author's lecture video on the chapter.
2. Read, understand, and run Java files "TestObservable.java", "MyObservable.java", "MyObserver.java", and "TestMyObservable.java".
3. Read and understand OCSF and Chat utility's source files "AbstractServer.java", "ConnectionToClient.java", "AbstractClient.java", "ChatClient.java", "ChatIF.java", "ClientConsole.java", and "EchoServer.java".
4. On Thursday, group 1 will present the design and logic of Java files "AbstractServer.java" and "ConnectionToClient.java", and group 2 will present the design and logic of Java files "AbstractClient.java", "ChatClient.java", "ChatIF.java", "ClientConsole.java", and "EchoServer.java".
5. Complete Project Exercise P3.2 on page 97 of the textbook, and post the zip file of your upgraded project containing all of its source files.
6. Each group should brainstorm on course project topic, and the group leaders should post its descriptions for further discussion. The project should be based on the OCSF framework and have user-friendly graphic user interfaces.
7. Discuss on the question: "What are the main advantages of separating user interface code from business-logic code?"

Week 5: Feb. 20 – Feb 26

1. Read Chapter 4 "Developing Requirements" of the textbook, and listen and watch the author's lecture video on the chapter.
2. Complete Project Exercise P3.3 on page 97 of the textbook to further enhance the Simple Chat client/server application.

3. Each group works out the *requirements document*, similar to that for the SimpleChat instance messaging program on pages 141-145 of the textbook, for the group course project; the group leaders will post the completed Word version of the document.
4. Each group plays out a brainstorming session for the course project as described on pages 122-123 of the textbook on Feb. 24. The group leader will be the *moderator* who is also responsible for coming up with one or two “trigger questions” for the session.
5. Discuss on the question: “What are the key factors for a successful brainstorming session?”

Week 6: Feb. 27 – Mar. 5

1. Continue unfinished tasks for Week 5.
2. Both teams conducted brainstorming sessions on their team projects, and presented their requirement analysis documents.

Week 7: Mar. 6 – Mar. 12

1. Read Chapter 5 “Modeling with Classes” of the textbook, and listen and watch the author’s lecture video on the chapter.
2. Both teams present their requirement analysis documents.
3. Study GUI and animation examples (TicTacToe.java, BounceBallApp.java, BallControl.java, and Ball.java; all are posted in Course Documents).
4. Both teams explore implementation strategies for team projects.

Week 8: Mar. 13 – Mar. 19

1. Read Chapter 5 “Modeling with Classes” of the textbook, and listen and watch the author’s lecture video on the chapter.
2. Both teams present on enhancement to the networked chat application.
3. Both teams work out the class diagrams for their course projects, and post them in Discussion Board.

Week 9: Mar. 27 – Apr. 2

1. Quiz 1 will be conducted on March 31. It will be open-book, concepts only, conducted through Blackboard. It covers textbook chapters 1 through 5.
2. Prototype 1 demo will be due on March 31. Both teams will present their first running prototype on March 31.
3. Read Chapter 6 “Using Design Patterns” of the textbook, and listen and watch the author’s lecture video on the chapter.
4. Discuss on the question: “Is the proposed CSIS academic integrity code fair?”

Week 10: Apr. 3 – Apr. 9

1. Continue reading Chapter 6 “Using Design Patterns” of the textbook, and listen and watch the author’s lecture video on the chapter.

2. Continue working on course projects as teams. Each team member must have clear assignment of his/her responsibilities for the next project presentation by April 14, 2005.

Week 11: Apr. 10 – Apr. 16

1. Read Chapter 7 “Focusing on users and their tasks” of the textbook, and listen and watch the author’s lecture video on the chapter.
2. Read Chapter 8 “Modeling interaction and behavior” of the textbook, and listen and watch the author’s lecture video on the chapter.
3. Continue working on course projects as teams. Each team will present its project and documentation on April 14, 2005.

Week 12: Apr. 17 – Apr. 23

1. Continue reading Chapter 8 “Modeling interaction and behavior” of the textbook, and listening and watching the author’s lecture video on the chapter.
2. Take Quiz 2 on April 21.
3. Continue to work on team projects.
4. Learn C# and Microsoft Visual Studio .NET

Week 13: Apr. 24 – Apr. 30

1. Read Chapter 9 “Architecting and designing software” and Chapter 10 “Testing and inspecting to ensure high quality”
2. Summary of software processes

Week 14: May 1 – May 7

1. Project demo and presentation continue
2. All project deliverables are due