University of Illinois at Urbana-Champaign
Department of Computer Science

Final Exam

CS 428 Software Engineering II
Spring 2012

May 8, 2012

TIME LIMIT = 3 hours
COVER PAGE + 16 PAGES

Upon receiving your exam, print your name and netid neatly in the space provided below; print your netid in the upper right corner of every page.

Name: _____
Netid: _____

This is a closed book, closed notes examination. You may not use calculators or any other electronic devices. Any sort of cheating on the examination will result in a zero grade.

We cannot give any clarifications about the exam questions during the test. If you are unsure of the meaning of a specific question, write down your assumptions and proceed to answer the question on that basis.

Answering **"I don't know"** (and nothing else) to any exam question is automatically worth 25% partial credit for that question.

Do all the problems in this booklet. The problems are of varying degrees of difficulty so please pace yourself carefully, and answer the questions in the order which best suits you. Answers to essay-type questions should be as brief as possible. **If the grader cannot understand your handwriting you may get 0 points**.

There are 13 questions on this exam and the maximum grade on this exam is **178 points**.

Question:	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
Points:	10	40	8	2	4	10	28	10	12	32	4	4	14	178
Score:														

UML Diagrams

1. Draw a state diagram describing a car at an intersection that follows the following rules: 10

    1. If the light is Red do not move.
    2. If the light is yellow, if speed is less than 10mph, stop, otherwise go.
    3. If the light is green
        - If the car desires to turn left
            (a) Yield until clear
            (b) Turn left
            (c) Repeat
        - If the car desires to go straight
            (a) Go
            (b) Repeat
        - If the car desires to turn right
            (a) Turn right
            (b) Repeat

2. PINGS is a two-dimensional sports game that simulates table tennis. Players control an in-game paddle by moving it across the screen. Players use the paddles to hit a ball back and forth. Points are lost when a player fails to return a ball. All players start with 5 points. Once a player has 0 points they are removed from the game.

   PINGS can have either 2 or 4 players. Every player has a name and color. One player is a server for the game, the others are clients. The clients connect over the internet. Players can play through different interfaces (e.g., web-broswer, iPhone, or Android app). After a game is completed the wining Players name is sent to the clients server where a leader board keeps track of the top winners of PINGS.

   (a) Create a UML Class Diagram for the PINGS game. To receive full credit, the UML Class Diagram must be able to provide all of the features mentioned above. [20]

   (b) Using the UML Class Diagram from part (a) create a sequence diagram that describes sequence of events for the scenario: a client joins a game created by the server, plays the game, wins, has their name submitted to the leaderboard. [20]

Continue your answer for question 2...

Use Cases

3. (a) Does the primary actor of a use case also need to be a stakeholder for it? Explain. [2]

   (b) Should the system implement at least one use case for which the stakeholder needs to be an actor? Explain. [2]

   (c) Describe two differences between *user stories* and *use case briefs*. [4]

Software Architecture and Patterns

4. Match each architectural pattern with the system that is typically implemented using this pattern: [2]

Architectural Patterns	**Applications**
a. Model-View-Controller	1. Compiler
b. Layered	2. Integrated Development Environment
c. Repository	3. Web Application
d. Pipe and Filter	4. TCP/IP

5. Relational Database Management System stores data and the relationships among the data are in tables. Give two advantages and two disadvantages of RDBMS over NoSQL solutions. [4]

Quality Assurance

6. (a) Software has both external and internal quality characteristics, e.g. usability and maintainability. Give two other external and two other internal characteristics, and briefly describe them. [4]

(b) For homework 4, you formally inspected someone elses code. Formal inspections differ from casual reviews such as code reviews between pair programmers. For example, a formal inspection requires a checklist (just as you have done for homework 4) that helps focus the reviewers attention on areas that have been problematic in the past. Give two other benefits of a formal inspection. You may give an example based on your experience with homework 4. [4]

(c) i. What is an Abstract Data Type? Give an example. [1]

ii. How does it differ from an object in Object Oriented Programming? [1]

Software Processes

7. (a) Describe examples for two properties of simple code according to the XP Simplicity Rules described in the Component Design lecture and at the XP Simplicity Rules wiki at c2.com. [4]

(b) i. Is XP just a special case of Crystal Clear? Explain. [3]

ii. Could a Scrum project be a special case of Crystal Clear? Explain. [3]

iii. Could a RUP project also be a Crystal Clear project? Explain. [3]

(c) In CS428 you have learned a number of different agile development processes, some of which you have followed in your project. Different processes have different characteristics, strengths and weaknesses. In the table given below, compare and contrast XP, Scrum and Crystal Clear for **three** categories. Select the categories from the ones listed below: **Testing, Roles, Activities, Values, Planning, Risk Management, Deliverables** 15

		Software Engineering Processes		
		XP	Scrum	Crystal Clear
Compare and Contrast Criteria				

Open Source

8. (a) It is said that releasing open source means more than just making your code freely available to the public. What else do you need to do in order to make the open-source project successful? List three characteristics of a successful open-source project. [3]

(b) How does Bazaar handle gathering requirements and planning? [2]

(c) Do bug reports in Bazaar and Cathedral differ? Explain. [2]

(d) What kind of project is not a good fit for being open source? Explain. [3]

Guest Lectures

9. (a) According to Robert C. Martin ("Uncle Bob"), in his lecture *Agility Craftsmanship and Ethics*, a programmers job involves more than just creating software that works. Describe one additional property he gives for well-crafted software. [2]

   (b) Robert C. Martin discusses the concept of short iterations in his lecture. [2]
   According to Mr. Martin, why is it desirable to have short iterations in a software development process? Describe one of the reasons he gives.

   (c) Eric Sink, in his lecture about distributed version control systems, describes UUIDs (Universally Unique Identifiers) as used in unique software defect identification. You may use examples in your answers.
      i. According to Mr. Sink, what is the main problem with UUIDs? [2]

      ii. What alternative does he give for uniquely identifying defects? [2]

   (d) Describe two advantages that distributed version control systems (DVCS) have over centralized version control systems, as given by Eric Sink in his lecture. [4]

User Interface

10. (a) What are two advantages and two disadvantages of separating user interface code from application code? [4]

(b) The concept of "safety" in user interfaces was discussed in class. Describe two examples of this concept using your own class project. If this does not apply to your class project, answer the same question for a different system. [4]

(c) Alan Cooper suggests that there are three dominant software interface paradigms: technology, metaphor, and idiomatic paradigms. Explain briefly what each paradigm is, and why Alan Cooper thinks that idiomatic paradigm is the best paradigm. [4]

(d) Give an example where the Command Pattern can be used in UI implementations, and draw a class diagram for your example. [4]

(e) In terms of user interface design,

i. What is the program model? [1]

ii. What is the user model? [1]

(f) Joel Spolsky says "A user interface is well-designed when the program behaves exactly how the user thought it would." Explain in your own words what he means by this statement, and give an example of a user interface that does not respect this principle. [3]

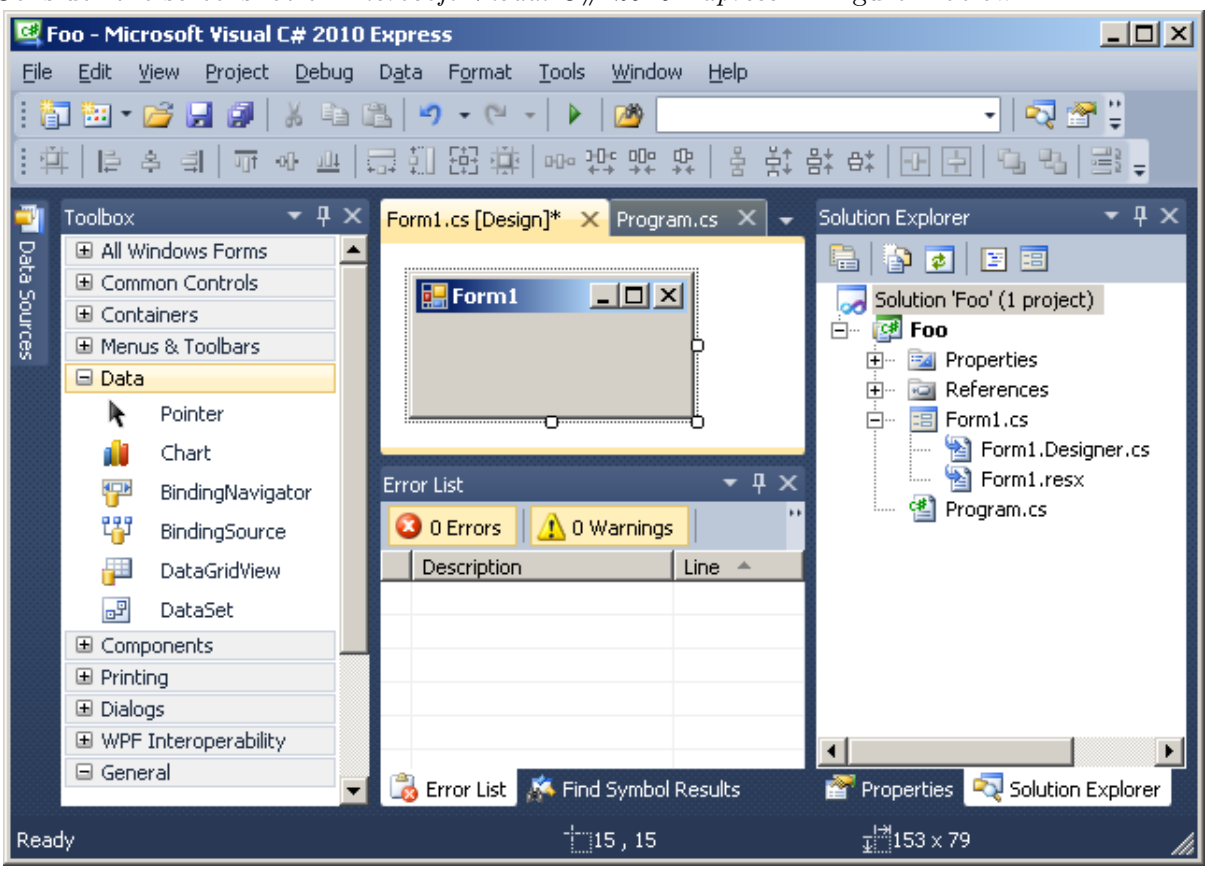
(g) Joel Spolsky suggests affordance and metaphors can help define user models.

i. Define affordance, and give an example from your own class project. [2]

ii. Define metaphor, and give an example from your own class project. [2]

iii. Joel Spolsky also advocates consistency in UI designs. Why is it important? [1]

(h) Consider the screenshot of *Microsoft Visual C# 2010 Express* in Figure 1 below.



i. Identify three design interface patterns that are clearly apparent/visible in the given screenshot. The bottom of the page contains possible pattern names. Label on Figure 1 the three design interface patterns you chose. [3]

ii. Briefly describe the benefit/reason of using each of the three patterns from part (i). [3]

Select from the following patterns:

Two-Panel Selector	One-Window Drilldown	Tabs (Card Stack)	Command History
Wizard	Extras On Demand	Intriguing Branches	Closable Panels
Simpleton Pattern	Action Panel	Smart Menu Items	Center Stage
Commander Pattern	Complete Interface	Progress Indicator	Virtual Environment
Multi-Level Undo			

Security

11. (a) Briefly describe how role-based access control enforces system security. [2]

(b) Briefly describe the principle of "least privilege" and give one benefit. [2]

Web Development

12. (a) Compare and contrast how are software updates distributed in cases when the client program is a standalone desktop app vs. cases when the client program runs in the web browser. [2]

(b) In terms of web-based system engineering, the phrase "content is king" refers to the fact that design factors have a little effect in the projects success if the content is boring, irrelevant or not appealing. Provide two ways to make the content relevant. [2]

Performance

13. Bazinga recently released their new game "Mighty Marsupial Madness"(MMM) to great success! But with growth comes headaches; Mighty Marsupial Madness is getting slower as more users sign up. Bazinga has hired you to determine areas where performance can be improved.

    MMM is a multiplayer game where players are broken up into groups of 20 players. Players can choose a group to join or Bazinga can randomly place a Player into a group. The different groups do not interact with one another until the round is over. Rounds typically last 15 minutes.

    You done some high level performance metrics and determined the following:

    - All servers are located in Seattle Washington.
    - There is an average of 20,000 players at any given time.
    - Each player is generating an average of 5 requests per second.
    - Bazinga has 5 servers, each with 16GB of memory and 50GB of harddisk.
    - These are intel i7 CPU machines running at 2.66Ghz
    - A typical game generates 5MB of data. Of that 5MB, 0.2 MB needs to be stored after the round has completed and needs to be globally accessible to all servers.
    - 20% of the different requests made make up 80% of the requests

    On the single-threaded client the render-update loop timing breakdown is:

    - 40% of time spent drawing graphics
    - 3% interpreting user input
    - 40% waiting for server responses
    - 17% sending a message to the servers

Questions for this section appear on the next page . . .

(a) List 3 changes that you propose to make the game more efficient. Your changes should either: improve performance, make the game use resources (e.g., bandwidth, hard-disk, CPU cycles) more efficiently or increase the perceived performance of MMM. [12]

(b) What is the Pareto Principle? Explain how this principle applies in the context of program optimization. [2]