

Student \_\_\_\_\_

Department of Computer Science and Engineering  
College of Engineering, University of Nevada, Reno

CS 425 / CS 625 Software Engineering

Midterm Test #2

November 21, 2005

**Test type:** Closed-book examination  
**Number of questions:** 13  
**Total points:** 29  
**Test weight:** 9%  
**Time:** 70 minutes  
**Notes:**

- For questions **1 to 9** indicate the correct answer (only one) on the answer sheets provided by the instructor. Each of the questions **1 to 9** has a one point value for a group total of **9** points.
- Questions **10 to 13** require that you elaborate your answers. You must also write these answers on the sheets provided by the instructor. The total group value of questions **10 to 13** is **20** points.

**Questions:**

- 1** *Control systems* are a type of real-time systems concerned with:
- Continuously checking and displaying the state of the system
  - Collecting sensor data for off-line processing and analysis
  - Continuously reading sensors and issuing commands to actuators
  - All of the above
- [1 point]
- 2** Which of the following is not a *generic architectural model for RTS* (real-time systems)?
- Data acquisition system
  - Version control systems
  - Monitoring system
  - Control system
- [1 point]
- 3** Which of the following is not a *primary style of user interaction*?
- Natural language
  - Code visualization
  - Direct manipulation
  - Menu selection
- [1 point]
- 4** Which of the following approaches can be used for *user-interface prototyping*?
- Script-driven prototyping
  - Internet-based prototyping
  - Visual programming languages
  - All of the above
- [1 point]

Student \_\_\_\_\_

- 5** Which of the following is not a technique for *user-interface evaluation*?
- Performing hierarchical task analysis
  - Surveying users by questionnaires
  - Instrumenting code to collect usage statistics
  - Observing users at work with the system and “thinking aloud”
- [1 point]
- 6** Three principles or practices of *extreme programming* are:
- Requirements formalization, simple design, small program releases
  - Refactoring, continuous risk analysis, small program releases
  - Collective ownership, pair programming, sustainable pace
  - All of the above (that is, each of the above lines contains three valid XP principles or practices)
- [1 point]
- 7** Two tools included in a *rapid application development (RAD) environment* are:
- Interface generator, report generator
  - Database programming language, links to office applications
  - Neither (a) nor (b)
  - Both (a) and (b)
- [1 point]
- 8** *Legacy systems* are:
- Old computer-based systems still in use by organizations
  - Software systems not yet implemented
  - Reverse engineered computer-based systems
  - None of the above
- [1 point]
- 9** Which of the following is a *strategic option* for dealing with *legacy systems*?
- Re-engineer the system
  - Scrap the system completely
  - Replace all or part of the system with a new system
  - All of the above
- [1 point]
- 10** Describe and compare the following two styles of user interaction: *direct manipulation* and *form fill-in*. Also, for each interaction style give a concrete example of software application that you are familiar with (briefly explain how it supports the specific interaction style).
- [5 points]
- 11** Briefly describe (2-3 lines each) three practices or principles (other than test-first development) that are used in *extreme programming (XP)*. In addition, describe in detail the practice of *test-first development*.
- [6 points]
- 12** Briefly discuss the following key factors that distinguish development and maintenance (2 to 4 lines each): *team stability*, *contractual responsibility*, and *program age and structure*. [3 points]
- 13** Consider your group project in CS425/625. Briefly describe the project concept (5 to 7 lines) and give examples of three functional requirements and two non-functional requirements. In addition, either (a) draw a sketch of your software product’s main user interface or (b) draw (part of) the use case diagram of your software product (with at least 5 use cases)
- [6 points]

Student \_\_\_\_\_