

Student \_\_\_\_\_

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

CS 425 / CS 625 Software Engineering

Midterm Test #2

November 17, 2004

**Test type:** Closed-book examination  
**Number of questions:** 13  
**Total points:** 30  
**Test weight:** 11%  
**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 **21** points.

**Questions:**

- 1** Which if the following is an *event-driven control model* used in architectural design?
- Manager model
  - Broadcast model
  - Call-return model
  - None of the above
- [1 point]
- 2** In object-oriented design *sequence models* show:
- The logical grouping of objects into subsystems
  - How objects change their state in response to events
  - The relationships between classes and their cardinalities
  - None of the above
- [1 point]
- 3** *Control systems* are a type of real-time systems that are concerned with:
- Collecting sensor data for off-line processing and analysis
  - Continuously reading sensors and issuing commands to actuators
  - Continuously checking and displaying the state of the system
  - All of the above
- [1 point]
- 4** Which of the following is not a *primary style of user interaction*?
- Forms fill-in
  - Natural language
  - Menu selection
  - GUI prototyping
- [1 point]

Student \_\_\_\_\_

- 5** Which of the following approaches can be used for *user-interface prototyping*?
- Internet-based prototyping
  - Script-based prototyping
  - Visual programming languages
  - All of the above
- [1 point]
- 6** Which of the following is not a technique for *user-interface evaluation*?
- Analyzing video recordings
  - Surveying users by questionnaires
  - Performing hierarchical task analysis
  - Instrumenting code to collect usage statistics
- [1 point]
- 7** Three principles or practices of *extreme programming* are:
- Refactoring, pair programming, incremental planning
  - Requirements formalization, simple design, small program releases
  - Test-first development, small program releases, continuous risk analysis
  - All of the above
- [1 point]
- 8** Two tools included in a *rapid application development (RAD) environment* are:
- Interface generator, report generator
  - Database programming language, links to office applications
  - Both (a) and (b) above
  - Neither (a) nor (b) above
- [1 point]
- 9** Which of the following software engineering techniques can help developing *fault-free software*?
- Quality management
  - Static verification
  - Safe programming
  - All of the above
- [1 point]
- 10** Describe the *interrupt-driven* control model used in architectural design. Also, indicate its advantages, disadvantages, and applicability.
- [5 points]
- 11** Faults in programs should be avoided, particularly when developing critical software systems. List *five error-prone programming constructs* and briefly describe how they may cause logical or run-time errors in software.
- [5 points]
- 12** Describe three key characteristics of *testing* in XP (*extreme programming*). Also, indicate three key characteristics of XP's *pair programming*. Comment on the advantages of these two XP practices.
- [6 points]
- 13** Indicate three guidelines for using colors in computer-user interfaces. Illustrate each guideline with a brief example from your own experience in interacting with computers where colors have a significant impact (positive or negative) on the quality of the user interface.
- [5 points]