

Student _____

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

CS 425 Software Engineering

Final Exam

December 14, 2009

Test type: Closed-book examination

Number of questions: 17 **Total points:** 48 **Weight:** 28% of course grade **Time:** 105 minutes

Notes:

- For questions **1 to 10** indicate the correct answer (only one) on the answer sheets provided by the instructors. Questions 1 to 10 have each a one point value for a group total of 10 points.
- Questions **11 to 17** require that you elaborate your answers. You must also write these answers on the sheets provided by the instructor. The total value of questions 11 to 17 is 38 points.

Questions:

- 1 Which of the following is a strategy in *risk planning*?
 - a. Minimization strategy
 - b. Avoidance strategy
 - c. Contingency strategy
 - d. All of the above[1 point]
- 2 The *critical path* in an activity network indicates:
 - a. The minimum time required to finish the project
 - b. The maximum time required to finish the project
 - c. The key milestones of the project
 - d. The number of tasks that can be performed in parallel[1 point]
- 3 Which of the following is a category of product-related *non-functional software requirements*?
 - a. Reliability requirements
 - b. Portability requirements
 - c. Both a and b
 - d. Neither a nor b[1 point]
- 4 Which of the following can be used for *requirements specification*?
 - a. Mathematical specifications
 - b. Structured natural language
 - c. Design description languages
 - d. All of the above[1 point]
- 5 Which of the following is a *centralized control model* used in architectural design?
 - a. Broadcast model
 - b. Distributed model
 - c. Manager model
 - d. Interrupt-driven model[1 point]
- 6 Which of the following is not a technique for *user-interface evaluation*?
 - a. Instrumenting code to collect usage statistics
 - b. Performing hierarchical task analysis
 - c. Surveying users by questionnaires
 - d. Analyzing video recordings of typical system use[1 point]

Student _____

- 7** Three principles or practices of *extreme programming* are:
- Refactoring, risk analysis, small program releases
 - Collective ownership, pair programming, sustainable pace
 - Requirements formalization, simple design, small program releases
 - All of the above (that is, each of the above lines contains three valid XP principles or practices) [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 core activity in the *user-interface design process*?
- Requirements engineering
 - UML modeling
 - System design
 - System prototyping [1 point]
- 10** The *cyclomatic complexity* of a program is equal to:
- The minimum number of test cases needed in path testing
 - The minimum number of test cases needed in partition testing
 - The minimum number of test cases needed in integration testing
 - The minimum number of test cases needed in performance testing [1 point]
- 11** Describe the *waterfall* software process model. Also, indicate its advantages, disadvantages, and applicability. [6 points]
- 12** Describe the *interrupt-driven model* used for control modeling in architectural design. Also, indicate its advantages, disadvantages, and applicability. [5 points]
- 13** Describe and compare the following two styles of user interaction: *direct manipulation* and *natural language*. Also, for each interaction style, give a concrete example of software application that you use or know about – briefly explain how it allows the specific interaction style. [5 points]
- 14** Briefly describe (2-4 lines each) five practices or principles that are used in *extreme programming* (XP). [6 points]
- 15** Concisely describe each of the following approaches for *test case design*: requirements-based testing, partition testing, and structural testing (3 to 5 lines each). [4 points]
- 16** Briefly discuss the following key factors that distinguish development and maintenance: *team stability*, *contractual responsibility*, and *program age and structure*. [4 points]
- 17** Consider your *group project* in CS425.
- Briefly describe the project's topic, utility, and most important features (6 to 10 lines).
 - List the project's three most important functional requirements.
 - Concisely summarize the project's implementation solution: operating platform(s), programming language(s) used, main subsystems/modules of the code (expectedly, between 4 and 6), and current status of the project.
 - Indicate three possible enhancements for your project. [8 points]