George Mason University

SWE 437 Software Testing and Maintenance J. Offutt

**Final Exam, in-class part 1 (A)**

Due 14 December, 1030pm

Submit written answers on this paper and pick up part 2 on your way out

Part 2 must be submitted through dropbox request by 1:15 on Wednesday

**Rules**:

1. You must answer these questions individually, without help from any person
2. You may not share this exam with anyone but yourself—doing so will be considered an honor code violation and, as a maximum, dismissal from the program
3. You may use your books, your notes, the course slides, and other resources on the web including on the book web site
4. Write neatly. I cannot grade what I can’t read, so if I can’t read your answer, I have to count it incorrect.

You probably have enough room to answer questions in the space provided. If you run short on space, you may staple a page of your own to the exam.

Name:    

Answer ten questions from the below list. Choose one sub-question from each of the 10 questions. Circle the number of the question that you answer. Each question is worth 1 point.

1. Use one of the following terms to fill in the blank for one of the three questions (1 point)

{ *decreased, failure, increased, had no change on, less, more, neither, related, unrelated* }

* 1. Web software led to \_\_\_\_\_\_\_\_\_\_ emphasis on software reliability
  2. Software quality is \_\_\_\_\_\_\_\_\_\_ to software security
  3. Agile processes have \_\_\_\_\_\_\_\_\_\_ software testing?

1. Answer one of two (1 point)
   1. Name **two** things that should be documented to support coding for change
   2. Name one advantage and one disadvantage, in terms of maintenance, of including more documentation
2. Answer one of two (1 point)
   1. Answer in **15 words or less**: What is the relationship between *evolutionary design* and *perfect out of the box*?
   2. Answer in **15 words or less**: Explain why programmers should ***not*** optimize their code by hand
3. Answer one of three (1 point)
   1. A Java method returned the wrong value, because of missing parentheses, causing the program to print the wrong value. From that description, what was the fault and what was the failure?
   2. Suppose a Java program has the expression “*A + B \* C*”, when it should have been “*(A + B) \* C*”. Give two sets of test values for the variables *A*, *B*, and *C*; one that would cause the expressions to have a different answer and one that would cause the expressions to have the same answer
   3. Answer in **15 words or less**: Explain the difference between a *fault* and a *failure*
4. Use one of the following terms to answer one of the three questions (1 point)

{ *controllability, coverage criterion, failure, fault, observability, test requirement, validation, verification* }

* 1. Evaluating software for compliance with intended usage
  2. Externally visible incorrect behavior
  3. A rule or collection of rules that impose test requirements on test sets

1. Answer one of three questions for the following JUnit test class (1 point)

|  |  |
| --- | --- |
| public class someTest {  private Set mySet = new HashSet();  @After public void tearDown() {  mySet = null;  }  @Test public void test1() {  mySet.add("microsoft");  assertEquals("[microsoft]", mySet.toString());  } | @Test public void test2() {  mySet.add("google");  mySet.add("facebook");  mySet.add("google");  assertEquals("[google, facebook, google]", mySet.toString());  }  @Test public void test3() {  assertEquals(**????**, mySet.toString());  }  } |

* 1. How many times will the *tearDown*() method be run?
  2. The oracle for *test3*() does not have a value. What is the correct value?
  3. Why doesn’t *test2*() work correctly?

1. Use one of the following terms to answer one of the three questions (1 point)

{ *happy path tests*, *invalid input tests*, *minimum viable product*, *refactoring*, *spike*, *test suite*, *user story* }

* 1. What does it mean to go through a period of intense programming to “get ahead” of the TDD tests?
  2. What type of tests are often **not** designed when designing TDD tests?
  3. What is changing the code to improve a **non-functional** quality, without changing the code’s behavior?

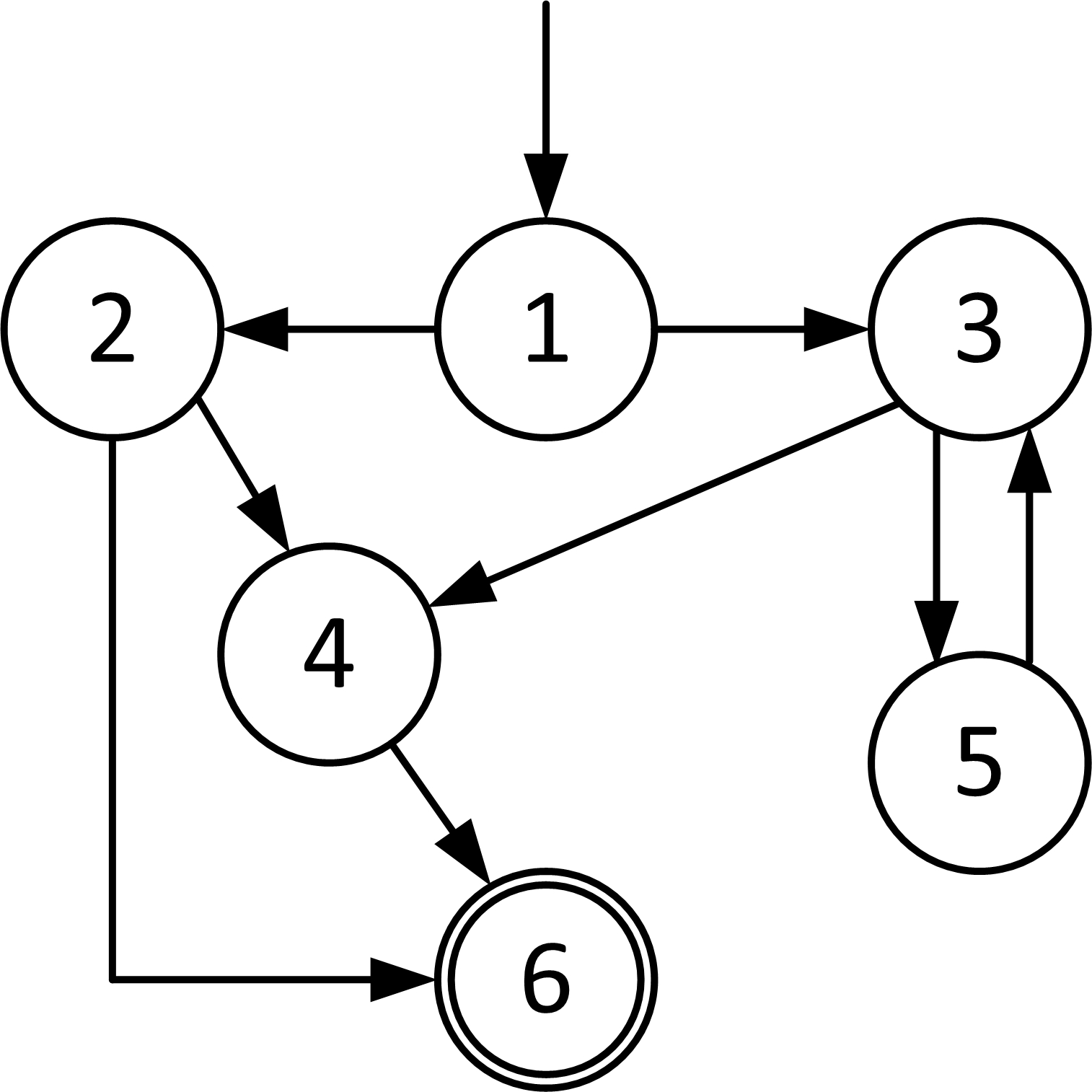
1. Answer one of three questions for ISP for the following test requirements (1 point)

The following table specifies three types of users for a home thermostat and the functions that each is allowed to perform.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Owners** | **Guests** | **Admins** |
| A. Set desired temp | **X** | **X** | **X** |
| B. Switch heat to cool | **X** | **X** | **X** |
| C. Enable remote monitoring (using wifi) | **X** |  |  |
| D. Change date & time | **X** |  | **X** |
| E. Connect to wifi | **X** |  |  |
| F. Run diagnostics |  |  | **X** |

* 1. List all test requirements for the test criterion C1: **test each function once**
  2. List all test requirements for the test criterion C2: **test one function for each user**
  3. If test criterion C3 says to **test every function for every user**, what coverage level can we achieve if we do not have the administrative password?

1. Answer one of two questions for the following graph (1 point)



* 1. List all the **prime paths** for this graph
  2. List all the **edge-pairs** for this graph

1. Answer one of three (1 point)
   1. Give truth assignments to satisfy CACC on the predicate: *(a & b & !c) | (b & c)*
   2. Give truth assignments to satisfy CACC on the predicate: *(a | !b | c) & (b | c)*
   3. Give truth assignments to satisfy CACC on the predicate: *(a & b) | (a & c) | (b & c)*