SE 2A04 Fall 2000

Midterm Test 2 Answer Key

Instructor: W. M. Farmer

You have 50 minutes to complete this test consisting of 4 pages and 13 questions. You may use your notes and textbooks. Circle the *best* answer for the multiple choice questions, and write the answer to the other questions in the space provided. Each answer you write should be either a technical expression or a sentence or phrase in English. If you answer a question with more than one sentence, only the first sentence will be read for marking; all others will be ignored. Good luck!

(1) [4 pts.] Two trees t_1 and t_2 *integer-overlap* if there is an integer k such that $k \in t_1$ and $k \in t_2$. (The term "tree" has the meaning defined in Lab Exercise 2.) It is possible for two trees to overlap but not to integer-overlap. Is this latter statement true or false?

- (a) True.
- (b) $\boxed{\text{False.}}$

(2) [4 pts.] The design document for a software product is a specification for how the product should be implemented.

- (a) $\boxed{\text{True.}}$
- (b) False.

(3) [4 pts.] The internal data structures of a module should be hidden from other modules.

- (a) $\boxed{\text{True.}}$
- (b) False.

(4) [4 pts.] The implementations of the interface functions of a module should be hidden from other modules.

- (a) $\boxed{\text{True.}}$
- (b) False.

(5) [5 pts.] What is the "secret" hidden by both of the tree modules for Lab Exercise 3?

Answer: How a tree is represented in the implementation of the module.

(6) [5 pts.] Suppose that a module is implemented as an object in an object-oriented programming language. What components of the object represent the interface of the module?

Answer: The public field and method declarations.

(7) [5 pts.] Which of the five software development activities (requirements, design, implementation, verification and analysis, and maintenance) involves writing documentation?

Answer: All five.

(8) [5 pts.] Which software life cycle model is the least effective when the problem to be solved is not well understood?

Answer: The waterfall model.

(9) [6 pts.] Let $\mathsf{Natural}$ and String denote the type of natural numbers and the type of ascii character strings, respectively. Also, let Stream denote the type of streams of natural numbers. The functions head and tail return the head and tail components, respectively, of a member of Stream. What are the types of head and tail?

Answer: $\mathsf{head} : \mathsf{Stream} \to \mathsf{Natural}$
$\mathsf{tail} : \mathsf{Stream} \to \mathsf{Stream}$.

(10) [6 pts.] Why do software engineers need to understand mathematics?

Answer: Software engineers need to use facts and techniques from mathematics to design and analyze software and need to use concepts and notation from mathematics to write precise software documentation.

(11) Suppose an ordinary global variable of type integer is represented as a module which is implemented as an object in an object-oriented programming language.

(a) [4 pts.] What fields would the object have?

Answer: A `data` field of type integer to hold an integer value.

(b) [4 pts.] What methods would the object have?

Answer: A `get` method for retrieving the integer value, and a `set` method for changing the integer value.

(c) [4 pts.] Which of the object's fields and methods should be private and which should be public?

Answer: `data` should be private, and `get` and `set` should be public.

(12) [20 pts.] Suppose that you are asked to design a finite state machine for recognizing the strings `abba` and `bb` in a stream of characters. You have described the set S of states, the set I of inputs, the set O of outputs, and the next state function `ns`; they are given below. The output `y` means `abba` or `bb` has been recognized, and `n` means neither has been recognized. You have not fully described the output function `out`; fill in the missing entries of the table for `out` given below.

$S = \{1, 2, 3, 4, 5\}$ and the initial state is 1.

$I = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

$O = \{\mathtt{y}, \mathtt{n}\}$.

ns	a	b	c
1	2	5	1
2	2	3	1
3	2	4	1
4	2	5	1
5	2	5	1

out	a	b	c
1	n	n	n
2	n	n	n
3	n	y	n
4	y	y	n
5	n	y	n

(13) [20 pts.] Consider a tape player with four buttons labelled reverse, play, forward, and stop/eject. The tape player responds to five events:

(a) Insert: a tape is inserted into the player.
(b) Reverse: the reverse button is pressed.
(c) Play: the play button is pressed.
(d) Forward: the forward button is pressed.
(e) Stop/Eject: the stop/eject button is pressed.

The behavior of the system is described by the following rules:

(a) A tape is stopped when it is first inserted into the player.
(b) A tape cannot be inserted when a tape is already in the player.
(c) Each button can be pressed at any time.
(d) Each button has no effect when there is no tape in the player.
(e) If there is a tape in the player, the tape is reversed, played, or forwarded when the reverse, play, or forward button, respectively, is pressed.
(f) If the tape is reversing, playing, or forwarding, the tape is stopped but not ejected when the stop/eject button is pressed.
(g) If there is a tape in the player that is stopped, the tape is ejected when the stop/eject button is pressed.

Model the behavior of the tape player as a finite state machine. Let each of the events be an input and assume that there are no outputs. There are five stable states:

1: No tape is in the player.
2: A tape is in and stopped.
3: A tape is in and reversing.
4: A tape is in and playing.
5: A tape is in and forwarding.

Below is an incomplete next state table for the finite state machine. Fill in the missing entries. If an input cannot happen in a certain state, put a "∗" meaning "don't care" in the corresponding entry in the table.

ns	Insert	Reverse	Play	Forward	Stop/Eject
1	2	1	1	1	1
2	∗	3	4	5	1
3	∗	3	4	5	2
4	∗	3	4	5	2
5	∗	3	4	5	2