NAME:_____

CSE331 Autumn 2010 Midterm Examination

November 10, 2010

- 50 minutes
- Open note, open book, closed neighbor, closed anything electronic (computers, web-enabled phones, etc.)
- An easier-to-read answer makes for a happier-to-give-partial-credit grader

	Possible points	Points
A. Specifications	**30**	
B. ADTs	**30**	
C. Miscellaneous	**30**	
Total	**90**	

Don't turn the page until the proctor gives the go ahead!

NAME:_____

A. Specifications

1) **(20 points) For each of the five specifications below, mark a + in the box if specification X (down the left column) is stronger than specification Y (across the top row), a – in the box if specification X is weaker than specification Y, an = in the box if they are equivalent, and a ● in the box if they are none of +, - or =. You only need to fill in the empty boxes in the lower-left triangular area.**

	$\alpha < \beta$	$\alpha \leq \beta$	$\alpha \leq \beta \mid \alpha < \beta$	$\alpha \leq \beta \ \& \ \alpha < \beta$	$\alpha \neq \beta$
$\alpha < \beta$	=				
$\alpha \leq \beta$		=			
$\alpha \leq \beta \mid \alpha < \beta$			=		
$\alpha \leq \beta \ \& \ \alpha < \beta$				=	
$\alpha \neq \beta$					=

2) **(5 points) True or false: Determining whether or not a Java subtype is a true subtype requires knowledge of the implementation of both the type and the subtype. Briefly explain (maximum of two sentences).**

3) **(5 points) Using implementation code as a specification is a poor idea. One reason is that it requires the client to read and understand all the code in the implementation before using it. In a maximum of two sentences, give one other reason why using implementation code is bad as an approach to specification.**

B. ADTs

Computers store multi-byte information either in big-endian or little-endian form. Consider an integer stored in two bytes (16 bits total). In a big-endian machine, the first byte represents the first eight bits of the integer and the second byte the second eight bits of the integer. These bytes are reversed on little-endian machines. So, in the example below, on a big-endian machine this would represent the integer $2^{11} + 2^9 + 2^2 + 2^0$ equaling 2048+512+4+1 = 2565. On a little-endian machine, this would represent the integer $2^{10} + 2^8 + 2^3 + 2^1$ equaling 1024+256+8+2 = 1290.

	First byte								Second byte							
Big-endian	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1
Little-endian																

1) (10 points) Is interpreting the numeric value of big- vs. little-endian two-byte integers more related to the notion of *abstraction function* or of *representation invariant*. Briefly justify (use a maximum of two sentences).

2) (15 points) Assume we are interested in manipulating 12-bit integers but choose to store them in two-bytes. Would the abstraction function have to change? If so, how? If not, why not? Separately, would the representation invariant have to change? If so, how? If not, why not?

NAME:_____

3) (5 points) Abstraction functions map concrete values to abstract values. In at most two sentences, explain why abstraction functions do not map abstract values to concrete values.

NAME:_____

C. Miscellaneous

1) **(15 points) Consider the following list of possible benefits of using immutable objects in Java – for each, is it true or false, providing one sentence of justification.**

   a. **hashCode can be determined at most once – that is, only when it is first actually requested by a client and then it can be cached**

   b. **It is easier to write immutable objects to disk.**

   c. **Only need to check a representation invariant after construction, but not again on the entry or exit from any other method in the class.**

   d. **If an immutable object throws an exception, it's never left in an undesirable or indeterminate state.**

   e. **Subclassing of a Java class that has only immutable objects is easier because pure typing is guaranteed in this situation.**

2) **(10 points) A common measure of the effectiveness of various white-box testing mechanisms is coverage: a set of tests, for example, that executes a higher percentage of statements in a program is considered to have higher coverage than one that executes fewer statements. True or false: A higher coverage measure implies that the implementation is more likely to satisfy the specification. In at most three sentences, justify your answer.**

3) **3) (5 points) True or false: A programming language in which a programmer cannot define new types does not need to allow a programmer to redefine `equality`. In at most two sentences, justify your answer.**