**GridWorld Case Study - Chapter 2 Quiz**

1. Consider the following circumstances for a bug.

I. The bug has a flower immediately in front of it.

II. The bug has a rock immediately in front of it.

III. The bug has a wall (grid boundary) immediately in front of it.

In which of these situations will the bug's canMove() method return false?

a. I only

b. II only

c. III only

d. I and II only

e. II and III only

2. Which of the following best describes how a bug turns when its turn() method is called?

a. 45 degrees to the left

b. 45 degrees to the right

c. 90 degrees to the left

d. 90 degrees to the right

e. 45 degrees randomly to the left or right

3 The reason there is no move method in BoxBug is...

a. A BoxBug never moves

b. The move method is inherited from Actor

c. The move method is inherited from BoxBug

d. The move method is inherited from Bug

e. The move method is hard-coded

4. Consider replacing the following code from BoxBug:

if (steps < sideLength && canMove())

with this code:

if (steps%sideLength != 0 && canMove())

How would this affect the boxbug's behavior?

a. The boxbug never turns, it continues in a straight line

b. The boxbug turns once after drawing one side, then continues in a straight line

c. There is no change in the boxbug pattern

d. The boxbug never moves, it only turns

e. The boxbug now makes a pattern like a Z

5. Consider the following MysteryBug that extends Bug. The code for MysteryBug is exactly the same as the code for BoxBug except that the following lines of code are placed at the beginning of the act method:

if(steps == sideLength)

sideLength--;

Which of the following best describes the behavior of a MysteryBug when the run button is clicked?

a. It forms a triangle.

b. It forms a square

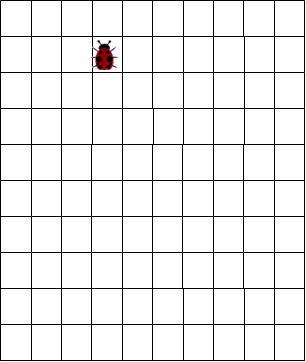
c. It forms a spiral with increasing sides; when the sides can no longer grow, it moves along the boundary of the grid.

d. It forms a spiral with smaller and smaller sides, when the side length reaches zero, then it forms a spiral with longer and longer sides.

e. It forms a spiral with smaller and smaller sides. When the side length reaches zero it does not move and turns in place.

6. Consider the following starting position for the Bug world. Which of the following lines in BugRunner would place

Bug chris = new Bug() in the indicated position?



a. world.add(2, 4, chris);

b. world.add(1, 3, chris);

c. world.add(new Location(2, 4), chris);

d. world.add(new Location(1, 3), chris);

e. world.getGrid().put(chris, 2, 4);

7. Consider the problem of having a bug draw a diamond pattern (a square tilted 45 degrees). Consider the following solutions.

I. Define and run the the class DiamondRunner that has the following code in its main method.

ActorWorld world = new ActorWorld();

BoxBug diamond = new BoxBug(3);

world.add(new Location(5, 1), diamond);

diamond.turn();

world.show();

II. Define the class DiamondBug that extends BoxBug and has the following constructor and no methods defined. DiamondRunner simply places a DiamondBug at location (5, 1).

public DiamondBug(int length)

{

super(length);

turn();

}

III. Define the class DiamondBug that extends BoxBug as shown below. DiamondRunner simply places a DiamondBug at location (5, 1).

public class DiamondBug extends BoxBug

{

private boolean dirNotSet;

public DiamondBug(int length)

{

super(length);

dirNotSet = true;

}

public void act()

{

if(dirNotSet)

{

turn();

dirNotSet = false;

}

super.act();

}

}

Which of these solutions produces a diamond pattern starting at location (5, 1) when the run button is clicked?

a. I only

b. II only

c. III only

d. I and II only

e. I, II, and III

8. Consider the following statements about the solutions in question 7.

I. Solution I is a good design because it uses the existing BoxBug class and needs minimal changes in code.

II. Solution II is a good design because it provides a diamond-specific bug that could be placed with the drop-down menus in the grid and work correctly, while making the simplest possible additions to its superclass.

III. Solution III is a good design because all movements of a bug should be restricted to its act method.

Which of these statements is accurate?

a. I only

b. II only

c. III only

d. I and II only

e. I, II, and III

9. Consider designing a FastBug class that extends the Bug class. A FastBug moves and turns following the same rules as a Bug except that when a FastBug invokes the act() method, it attempts to move like a bug twice instead of once (turning if blocked each time). Consider the following implementations of the act() method for the FastBug class.

I.

public void act()

{

if(canMove())

move();

else

turn();

if(canMove())

move();

else

turn();

}

II.

public void act()

{

super.act();

super.act();

}

Which of the following statements about these implementations is most accurate?

a. Implementation I works as intended but implementation II does not because super.act() cannot be called twice

b. Implementation II works as intended, but implementation I does not.

c. Both implementations works as intended and are equally good designs

d. Both implementations work as intended but implementation I is the better design because it has less dependency on its super class

e. Both implementations work as intended, but implementation II is better because it does not repeat code from its superclass

10. Suppose a new BoxBug(3) has been created facing north at location (3,3) in grid that has five rows and five columns and no other actors. When the run button is clicked, how often will this box-bug visit the location (2, 3)?

a. 0

b. 1

c. 2

d. 3

e. An infinite number of times