**GridWorld Case Study - Chapter 3 Quiz**

1. Assume the following statements have been executed.

Location loc1 = new Location(1,2);

Location loc2 = new Location(2,3);

Consider the expression

dir == loc2.getDirectionToward(loc1)

Which of the following replacements for "dir" will make this expression true?

a. Location.SOUTHEAST

b. Location.NORTHWEST

c. Location.WEST

d. Direction.NORTHWEST

e. No replacement will make the expression true because directions should be compared using ".equals(...)" rather than "=="

2. Make the same assumptions as in the previous question.

What will be returned by the call

loc2.getAdjacentLocation(135) ?

a. Location (1,2)

b. Location (1,4)

c. Location (3,2)

d. Location (3,4)

e. Nothing will be returned; the call will not compile because the parameter is the wrong type

3. Assume that Grid<Actor> grd has been initialized as a bounded grid with 10 rows and 20 columns.

What will be returned by the call

grd.isValid(new Location(5, 20)) ?

a. true

b. false

c. "yes"

d. "no"

e. Nothing, The statement will not compile because the type of the parameter should be Actor, not Location

4. Make the same assumption about grd as in the previous question.

Assume the following statements have been executed.

Actor a1 = new Actor(); Actor a2 = new Actor(); Actor a3 = new Actor();

Actor a4 = new Actor(); Actor a5 = new Actor();

grd.put(new Location(2,2), a1); grd.put(new Location(2,3), a2);

grd.put(new Location(3,3), a3); grd.put(new Location(4,3), a4);

grd.put(new Location(4,4), a5);

What is returned by the call

grd.put(new Location(3,3), new Actor()) ?

a. Nothing is returned. It is a void method

b. Nothing is returned because the parameters are incorrect; the call should be ............. grd.put(3,3, new Actor())

c. null

d. location (3,3)

e. a3

5. Make the same assumptions about grd and the actors placed in it as for the previous question.

What is returned by the call

grd.getValidAdjacentLocations(new Location(0,0)) ?

a. The list of locations {(0,1), (1,0)}

b. The list of locations {(0,1), (1,1), (1,0)}

c. The list of locations {(-1, 0), (0, -1), (0,1), (1,0)}

d. The list of locations {(-1, -1), (-1, 0), (-1, 1), (0, -1), (0,1), (1, -1), (1,0), (1,1)}

e. Nothing is returned. The statement will not compile because the type of parameter should be Actor, not Location.

6. Make the same assumptions about grd and the actors that have been placed in it as for the previous two questions.

What is returned by the call

grd.getEmptyAdjacentLocations(new Location(3,3)) ?

a. The list of locations {(2,2), (2,3), (2,4), (3,2), (3,4), (4,2), (4,3), (4,4)}

b. The list of locations { (2,3), (3,2), (3,4), (4,3)}

c. The list of locations {(2,4), (3,4), (4,2), (3,2)}

d. The list of locations {(3,4), (3,2)}

e. Nothing is returned. The statement will not compile because the type of parameter should be Actor, not Location

7. Assume the same assumptions about the initialization of grd and several actors as in the previous three questions.

What is returned by the call

grd.getNeighbors(a3) ?

a. The list of locations {(2,3), (3,2), (3,4), (4,3)}

b. The list of locations {(2,2), (2,3), (2,4), (3,2), (3,4), (4,2), (4,3), (4,4)}

c. The list of locations {(2,2), (2,3), (4,4), (4,3)}

d. The list of actors {a1, a2, a5, a4}

e. Nothing because the statement will not compile because the parameter should have type Location, not Actor

8. Assume that several actors have been placed in a bounded 10x10 Grid<Actor> grd. The method numNeighbors(loc)is intended to return the number of actors that are adjacent to loc. A partial definition for numNeighbors is given below.

// precondition: loc is valid in grd

// return: the number of actors in locations adjacent to loc in grd.

public int numNeighbors(Location loc)

{

return // missing code

}

Consider the following replacements for // missing code.

I. grd.getEmptyAdjacentLocations(loc).size();

II. grd.getOccupiedAdjacentLocations(loc).size();

III. grd.getNeighbors(loc).size();

Which of these replacements for missing code makes the method numNeighbors work as intended?

a. II only

b. III only

c. I and II only

d. II and III only

e. I, II, and III