

**I. Answer True or False**

- 1) The line `for (int i = 0; i < Location.FULL_CIRCLE / Location.HALF_CIRCLE, i++)` and the line `for (int i = 0; i < 8, i++)` are equivalent. 1) \_\_\_\_\_
- 2) ChameleonCritters change color every time they act. 2) \_\_\_\_\_
- 3) Critters never change direction when they do move. 3) \_\_\_\_\_
- 4) ChameleonCritters never change direction when they do move. 4) \_\_\_\_\_

**II. Answer whether the following instantiation statements are Valid or Invalid**

- 1) `Rock r1 = new Rock( );` 1) \_\_\_\_\_
- 2) `Bug b2 = new Bug(Color.BLUE);` 2) \_\_\_\_\_
- 3) `BoxBug bb3 = new BoxBug( );` 3) \_\_\_\_\_
- 4) `Critter c4 = new ChameleonCritter( );` 4) \_\_\_\_\_
- 5) `Actor a5 = new Actor( );`  
`Flower f5 = new Flower(a5.getColor( ));` 5) \_\_\_\_\_

**III. Answer the following**

- 1) Given the following code below, what is stored in `loc2` ? 1) \_\_\_\_\_  
`Location loc1 = new Location(1, 1);`  
`Location loc2 = loc1.getAdjacencntLocation(260);`
- 2) If a Bug, `bug2`, is facing WEST then the code `setDirection(bug2.getDirection( ) + Location.HALF_RIGHT);` will be what direction? 2) \_\_\_\_\_

**over**

IV. Multiple Choice

Use the code below to answer #1 below. The getAdjacentLocations method of the AbstractGrid class is modified as shown below. Changes are in **BOLDFACE**.

```
public ArrayList<Location> getValidAdjacentLocations(Location loc)
{
    ArrayList< Location> locs = new ArrayList<Location>;
    int d = Location.NORTH;
    for (int i = 0; i < Location.FULL_CIRCLE / Location.HALF_RIGHT; i++)
    {
        if (i % 2 == 0)
        {
            Location neighborLoc = loc.getAdjacentLocation(d);
            if (isValid(neighborLoc))
                locs.add(neighborLoc);
        }
        d = d + Location.HALF_RIGHT;
    }
    return locs;
}
```

- 1) What is the effect of this change? 1) \_\_\_\_\_
- a) The method returns all valid locations whose direction from loc is an odd number.
  - b) The method returns all valid locations whose direction from loc is an even number.
  - c) The method returns all valid adjacent locations that share a side, but not a corner point, with loc.
  - d) The method returns all valid adjacent locations that share a corner point, but not a side, with loc.
  - e) The method is equivalent to the original, and returns all valid adjacent locations.
- 2) To put a Bug into a Grid gr, with Location loc = new Location(1, 1); which of the following would accomplish this so that the Bug would know its location in the grid. 2) \_\_\_\_\_
- a) Bug b2 = new Bug( );  
b2.putSelfInGrid( );
  - b) Bug b2 = new Bug( );  
b2.putSelfInGrid(gr);
  - c) Bug b2 = new Bug( );  
b2.putSelfInGrid(loc);
  - d) Bug b2 = new Bug( );  
b2.putSelfInGrid(gr, loc);
  - e) Bug b2 = new Bug( );  
gr.put(loc,b2);

**continued**

V. In a 10 x 10 bounded Grid gr there are three Critters each facing NORTH and a Rock: critter1 at (0, 0), critter2 at (0,1), critter3 at (1, 1), rock1 at (0, 2) and no other actors in the grid. Assume that the ArrayList add method will correctly put objects into the ArrayLists below and that all possible grid cells are checked appropriately.

1) What will be stored in ArrayList<Actor>, which stores the values returned by the method:

a) gr.getNeighbors(critter1.getLocation( ))

1a) \_\_\_\_\_

b) gr.getNeighbors(critter2.getLocation( ))

1b) \_\_\_\_\_

c) gr.getNeighbors(critter3.getLocation( ))

1c) \_\_\_\_\_

2) What will be stored in ArrayList<Location>, which stores the values returned by the method:

a) gr.getValidAdjacentLocations(critter1.getLocation( ))

2a) \_\_\_\_\_

b) gr.getValidAdjacentLocations(critter2.getLocation( ))

2b) \_\_\_\_\_

c) gr.getValidAdjacentLocations(critter3.getLocation( ))

2c) \_\_\_\_\_

3) What will be stored in ArrayList<Location>, which stores the values returned by the method:

a) gr.getOccupiedAdjacentLocations(critter1.getLocation( ))

3a) \_\_\_\_\_

b) gr.getOccupiedAdjacentLocations(critter2.getLocation( ))

3b) \_\_\_\_\_

c) gr.getOccupiedAdjacentLocations(critter3.getLocation( ))

3c) \_\_\_\_\_

4) What value is: critter2.getActors( ).size( ) ?

4) \_\_\_\_\_

5) What value is:

a) critter1.getMoveLocations( ).size( ) ?

5a) \_\_\_\_\_

b) critter2.getMoveLocations( ).size( ) ?

b) \_\_\_\_\_

c) critter3.getMoveLocations( ).size( ) ?

c) \_\_\_\_\_