

Set3

1. How would you access the row value for loc1?

```
Location loc1 = new Location(4, 3);
```

```
Location loc2 = new Location(3, 4);
```

Answer: loc1.getRow()

2. What is the value of b after the following statement is executed?

Answer: False

3. What is the value of loc3 after the following statement is executed?

```
Location loc3 = loc2.getAdjacentLocation(Location.SOUTH);
```

Answer: (4, 4)

4. What is the value of dir after the following statement is executed?

```
int dir = loc1.getDirectionToward(new Location(6, 5));
```

Answer: (6,5)

5. How does the getAdjacentLocation method know which adjacent

location to return?

Answer: the row values increase as you go south (down the screen) and the column values increase as you go east (to the right on the screen), of course, the row value decreases as you go north and the column values decreases as you go west.

SET4

1. How can you obtain a count of the objects in a grid? How can you obtain a count of the empty locations in a bounded grid?

Answer: `ArrayList<Location> getOccupiedLocations();`

2. How can you check if location (10,10) is in a grid?

Answer: `Boolean isValid(10,10);`

3. Grid contains method declarations, but no code is supplied in the methods. Why? Where can you find the implementations of these methods?

Answer: The implements of these methods are in the `BoundedGrid` as well as `UnboundedGrid`

4. All methods that return multiple objects return them in an `ArrayList`. Do

Answer: you think it would be a better design to return the objects in an array? Explain your answer.

Answer: No, I don't think so. ArrayList can be more convenient since it can help objects to be inserted into the List more easily.

SET5

1. Name three properties of every actor.

Answer: Color, direction and location

2. When an actor is constructed, what is its direction and color?

Answer: The default color is blue

The default direction is north

3. Why do you think that the Actor class was created as a class instead of an interface?

Answer: I think a class is more easily to extends

4. Can an actor put itself into a grid twice without first removing itself?

Can an actor remove itself from a grid twice? Can an actor be placed into

a grid, remove itself, and then put itself back? Try it out. What happens?

Answer: When I put the bug into a grid twice without first removing itself, the program throw out an exception which says:” This actor is already contained in a grid”

An actor can not remove itself from a grid twice

If an actor be placed into a grid and then remove itself, it can put itself back

5. How can an actor turn 90 degrees to the right?

Answer: It has to turn for twice.

SET6

1. Which statement(s) in the canMove method ensures that a bug does not try to move out of its grid?

Answer: `Location next = loc.getAdjacentLocation(getDirection());`

`if (!gr.isValid(next))`

`return false;`

2. Which statement(s) in the canMove method determines that a bug will not walk into a rock?

Answer: `Actor neighbor = gr.get(next);`

```
return (neighbor == null) || (neighbor instanceof Flower);
```

3. Which methods of the Grid interface are invoked by the canMove method and why?

Answer: isValid(); It is used to determine whether the next location the bug to go is valid or not.

4. Which method of the Location class is invoked by the canMove method and why?

Answer: getAdjacentLocation(getDirection())

It is used to get the next location that the bug will reach if the bug go down straight.

5. Which methods inherited from the Actor class are invoked in the canMove method?

Answer:

getGrid()

getLocation()

getDirection()

6. What happens in the move method when the location immediately in

front of the bug is out of the grid?

Answer: It will remove it self from grid

7. Is the variable loc needed in the move method, or could it be avoided by calling getLocation() multiple times?

Answer: The variable loc is needed in the move method, it can't be avoided by calling getLocation() multiple times because when the bugs move, location of the bug will change. However, what we want is to put a flower behind the bug everytime when the bug move.

8. Why do you think the flowers that are dropped by a bug have the same color as the bug?

Answer: No, the color of the flowers will change as time goes by.

9. When a bug removes itself from the grid, will it place a flower into its previous location?

Answer: No, it won't

10. Which statement(s) in the move method places the flower into the grid at the bug's previous location?

Answer: `Flower flower = new Flower(getColor());`

11. If a bug needs to turn 180 degrees, how many times should it call the turn method?

Answer : 4 times