

**Set 3, Page 19**

The API for the Location class is in Appendix B.

Assume the following statements when answering the following questions

```
Location loc1 = new Location(4, 3);  
Location loc2 = new Location(3, 4);
```

1. How would you access the row value for loc1?
2. What is the value of b after the following statement is executed?
3. What is the value of loc3 after the following statement is executed?

```
boolean b = loc1.equals(loc2);
```

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

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

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

5. How does the getAdjacentLocation method know which adjacent location to return?



**Set 5, Page 23**

The API for the `Actor` class is in Appendix B.

1. Name three properties of every actor.

(a)

(b)

(c)

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

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

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

(b) Can an actor remove itself from a grid twice?

(c) Can an actor be placed into a grid, remove itself, and then put itself back? Try it out. What happens?

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

**Set 6, Page 25**

The API for the `Bug` class is in Appendix C.

1. Which statement(s) in the `canMove` method ensures that a bug does not try to move out of its grid?
2. Which statement(s) in the `canMove` method determines that a bug will not walk into a rock?
3. Which methods of the `Grid` interface are invoked by the `canMove` method and why?
4. Which method of the `Location` class is invoked by the `canMove` method and why?
5. Which methods inherited from the `Actor` class are invoked in the `canMove` method?
6. What happens in the `move` method when the location immediately in front of the bug is out of the grid?
7. Is the variable `loc` needed in the `move` method, or could it be avoided by calling `getLocation()` multiple times?
8. Why do you think the flowers that are dropped by a bug have the same color as the bug?
9. When a bug removes itself from the grid, will it place a flower into its previous location?
10. Which statement(s) in the `move` method places the flower into the grid at the bug's previous location?
11. If a bug needs to turn 180 degrees, how many times should it call the `turn` method?