|  |  |  |
| --- | --- | --- |
| **APCS Exposure Java** | **GWExercises05** | **Date:** |
| **Name:** | | **Period:** |

**Execute the GridWorldLab01 project, click RUN and observe the grid display.**

1. What do you see when you click on an occupied cell?

2. What do you see when you click on an empty cell?

3. How many different classes are available with **GridWorldLab05**?

4. How many classes do you see in **Figure 03**?

5. How many objects do you see in **Figure 03**?

6. How many classes do you see in **Figure 11**?

7. How many objects do you see in **Figure 11**?

**Refer to Step 07 for the next few questions.**

8. What do the **BoxBug**, **ChameleonCritter**, **CrabCritter**, **OctagonBug** and the **SpiralBall** classes all have something in common in terms of the location of their **.java** files?

9. Refer to the previous question.

How are the **Actor**, **Bug** and **Critter** classes different?

10. How can you add new objects to the grid during program execution?

11. Refer to the previous question.

Some of the class names are followed by an empty set of parentheses. Some other classes display data types between the parentheses. What is the difference between these?

12. Refer to the previous question.

If you want to add a **Bug** object, you will notice it has both one constructor with empty parentheses and another one with something inside the parentheses. What is the difference between these 2 constructors in terms of what they do?

13. Refer to the previous question.

What are the names of these 2 different *types* of constructors?

14. Can it be said that the **Flower** class *overloads* the constructors like the **Bug** class?

15. How can you remove existing objects from the grid during program execution?

16. What is the behavior of the **Critter** object?

17. What is the behavior of the **ChameleonCritter** object?

18. How do **CrabCritter** objects **move**?

19. Does a **CrabCritter** object ever **turn**? If so when and how?

20. Both **Critter** objects **CrabCritter** objects can eat other objects, but there is a difference.

Look closely at the execution and explain this difference.

21. Refer to the previous question.

Can **Critters** and **CrabCritters** eat any kind of objects or are there certain objects they cannot eat?

If so, what are they?

**Refer to Step 09b for the next few questions.**

22. In the **Class View** window, what do *blue rectangles*indicate?

23. In the **Class View** window, what do *pink rectangles*indicate?

24. In the **Class View** window, what does a *lock* on a *pink/blue rectangle*indicate?

**Double click on OctagonBug.java file in the File View window.**

**DO NOT SCROLL DOWN PAST THE COMMENTS!**

25. List the *attributes* of the **OctagonBug** class.

26. List the *methods* of the **OctagonBug** class.

27. List the **private** members of the **OctagonBug** class.