Grade 12 Assignment #5 Java Programming

Build Your Own Android

In this project you will create a simple Droid (robot) that can be activated, charged, and hover above ground. This project creates some baseline behaviors of the droid.

Upon completion, feel free to explore more and add additional behavior(s) to your Droid.

Tasks:

There are 34 Tasks to complete for this assignment:

1. Open the BlueJ IDE editor to create the code needed for the program called Build Your Own Android.

2. First set up the BlueJ IDE. Create a project (file) with a public class called Droid.

3. Create an int instance variable called batteryLevel. Do not set it equal to anything at the moment.

This is a global variable. All global variables should be kept together for ease of understanding the code. Global variables are kept at the top of the code underneath the public class declaration.

4. Inside of the public class called Droid, underneath the global variables, create a Droid constructor. Inside of the Droid constructor, set batteryLevel equal to 100. Now every time a Droid is created, its battery level will be at 100 percent.

5. Next, create a method called activate. The method should not return any data type.

6. Inside of activate, print out a message to the user that lets them know their Droid is activated and alive. For example: Activated. How can I help you?.

7. All Droid actions should use up some battery life. On the next line, decrease the battery level by 5 percent.

   Hint: batteryLevel is equal to batteryLevel minus 5.

8. We should always update the user on the battery level every time the Droid performs an action. On the next line, print out the battery level.

   Hint: System.out.println("Battery level is: " + batteryLevel + " percent.");.

9. Let's provide a way for users to charge their Droid. Create a method called chargeBattery. It should accept an int parameter called hours. The method should not return any data type.

10. The first thing the method should do is inform the user that their droid is charging. Print out something like: Droid charging....

11. Next, the battery level should increase by the number of hours specified by the user. But we need to convert the hours into a percentage. To charge the entire battery takes about 5 hours. That means that each hour charges the battery by about 20%.

    Hint: 1 hour = 20% battery life →

    X hours * 20 percent/hour = Percent (increase in battery life)

    Hint: batteryLevel is equal to batteryLevel plus percent.

12. We have to make sure the user doesn't overcharge the Droid. Inside of chargeBattery, write an if statement that checks if the battery level is greater than 100.

13. If the battery level is greater than 100, set the battery level equal to 100, then print out the battery level to the user on the next line (refer to step 8 if you need help).

14. Otherwise, in an else block, simply print out the new value of the battery level to the user (refer to step 8 if you need help).

15. Next, create a method called checkBatteryLevel. It should return an int.

16. Inside of the method, reduce the batteryLevel by 5 and print out the battery level to the user.

17. On the next line, return the battery level.

18. Create another method called hover. The method should not return a value. It should accept an int parameter called feet.

19. This method will instruct the Droid to hover above the ground, but we can't let the Droid hover too far off the ground. Inside of the method, use an if statement to check if feet is greater than 2.

20. Inside of the if statement, print out a friendly error to the user, for example: Error! I cannot hover above 2 feet. Then set the hovering height to 2 feet and inform the user that the hovering height has been set to its maximum (2 feet).

21. Otherwise, in an else block, print out to the user a statement to the user like: Hovering....

22. On the next line, you have to decide how much to decrease the battery level. Use a switch statement to choose between the inputted int variable feet. For the case that the droid is hovering just above the floor (feet = 0) it requires 15% battery life. The battery life required is 20% for the case of hovering 1 foot above the floor. While the max height requires 30% battery life.

23. On the next line, print out the battery level to the user.

24. Next, create a main method. This is the main body of programming code that is executed when run. Note: the main method must be defined *exactly the same way* every time it is created. Refer back to the lesson if you need to review the main method.

    In this assignment, the main part of the code is extremely simple. It will focus on creating an object that belongs to the class through the class constructor. Once an object is created, that object can call the methods (specialized functions) that belong to the class. The main part of the code will simply call some of those methods (activate, hover, chargebattery, …) and print out the results to the screen.

25. Inside of the main method, create a Droid object.

26. On the next line, call the activate method on the object.

27. Next, call the chargeBattery method and specify 5 as the hours parameter.

28. Next, call the hover method and specify a 3 feet parameter.

29. Then, call the hover method again and specify a 1 foot parameter.

30. Next, call the checkBatteryLevel method to make sure you have enough power to keep playing.

31. Lastly, call the chargeBattery method for an hour.

If you completed the project correctly, the output will indicate first a battery level of 95, then 100 with a comment about being fully charged, then 70 with an error about exceeding the max hovering height, 50, 45, 65.

Feel free to explore more with the program. What are some ways in which the program could be improved?

For example, make sure to include enough printed out statements to the screen (System.out.println()) so that the user knows what is going on and that they are clear regarding the information they are receiving. Make sure there are enough carriage return println() statements to separate all the comments being printed out to the screen otherwise it will be hard to read.

32. It would be helpful to describe to other developers what this small Java program does. Write some comments that describes what this program does.

    a. Use multi-line comments to (/* comment in the middle of */):
        i. Write one at the top of the code (before the public class Continents designation) that gives a quick intro/description the assignment.
        ii. Write one at the top of the code (before the public static void main(String[] args)) that summarizes the program.
    b. Use a single line comment to (//then comment):
        i. Create 3 comments anywhere you deem necessary or important in the code. Remember the comment is to highlight or explain what is going on or what is being done in a particular way or used and why.
        ii. Identify each of the Methods created by indicating its purpose.
        iii. Identify the last line of code (anything that ends with a curly bracket }) in every function by writing “End of BLAH-BLAH function”.

iv. Identify the end of the program.

33. Once your program is complete, make sure to test it using the BlueJ IDE (do not submit a program that does not work).

34. Lastly, upload (drag and drop) your assignment to the portal. Look for your name under the Assignment 5 webpage.