Final Performance Task of the Year
(Worth 3 Times Weight)

You will develop (using Blue Jay) a program with two classes.

The AlarmClock Class with construct an alarm clock object with all the attributes of an Alarm Clock. It should include (obviously) a constructor method as well as getter and setter methods. A toString method and any other methods you can develop to make the Class a robust program.

The testAlarmClock class will have a main method that demonstrates your AlarmClock Class by creating instances and output when ran.

To submit this, you will need to zip the file that contains this program. It should contain NO other programs. You will be allowed to submit this only ONCE to my school email. I will email you a confirmation that I received the file. So make sure you test your code very well.

DUE 12/21/17 At 3PM ***or earlier.*** **NO WORK ACCEPTED PAST 12/21/17 3PM Email Date Stamp.**

Rubric

100	Demonstrated All methods in the AlarmClock Class.	Methods developed & demonstrated, reflect EVERYTHING an Alarm Clock could do in real life.	The output generated clearly explained what was taking place during the run. Code is clearly documented with comments.
90	Demonstrated ALMOST all methods in the AlarmClock Class.	Methods developed & demonstrated, reflect ALMOST everything an Alarm Clock could do in real life.	The output generated clearly explained what was taking place during the run. Code is almost all documented with comments.
80	Demonstrated MOST all methods in the AlarmClock Class.	Methods developed & demonstrated, reflect MOST of what an Alarm Clock could do in real life.	The output generated does a good job explaining what was taking place during the run. Code is mostly all documented with comments.
70	Demonstrated the basic methods in the AlarmClock Class.	Methods developed & demonstrated, reflect the BASICS of what an Alarm Clock could do in real life.	The output generated does a fair job explaining what was taking place during the run. The grader may have some minor confusion of what is taking place. Code is somewhat documented with comments.
60	Demonstrated 1 basic method in the AlarmClock Class.	Methods developed & demonstrated, reflect the less than the BASICS of what an Alarm Clock could do in real life.	The output generated is confusing as to what was taking place during the run. Code is minimally documented with comments.
50	Demonstrated 0 basic methods in the AlarmClock Class. Or Nothing was submitted.	Methods developed & demonstrated, reflected nothing of what an Alarm Clock could do in real life. Or Nothing was submitted.	The output generated (if any) is very confusing as to what was taking place during the run. Code is minimally (if at all) documented with comments. Or Nothing was submitted.