**Problem**:

You are to write a program that will ask the user for a grade and give them options for applying a curve to that grade. It should give them the option to curve by 10 points or 10 percent. It should also give them the option to curve the grade by the number of points that they enter or a percentage that they enter. It should use a menu to allow the user to select from the different options for curving the grade.

**Algorithm**

* Import Scanner
* Import DecimalFormat
* Create new Scanner object
* Declare variables and constants
* Clear the screen
* Prompt user for grade
* Display grade adjuster menu
* Ask user for menu selection

**Calculations**:

* Begin Switch Statement by checking what the user entered for the menu selection
* If 1, apply a 10 point curve by adding 10 to the original grade
* If 2, apply 10% curve by multiplying the original grade by 10%, then add that to the original grade.
* If 3, add the number of points entered by user in the getCurvePoints method to the original grade
* If 4, apply curve of original points\*percentage entered by user in the getCurvePercentage method to the original grade
* If 5, exit program
* If invalid selection, display error message

**Method**: inputGrade

**UML Entry**: + inputGrade():double

**Algorithm**: Create scanner, then prompt user to enter original grade, then return what was entered

**Method**: displayMenu

**UML Entry**: + displayMenu():void

**Algorithm**: Print out menu selections

**Method**: freezeScreen

**UML Entry**: + freezeScreen():void

**Algorithm**: Create scanner. Tell user to press enter. Freeze the screen until they press enter.

**Method**: clearScreen

**UML Entry**: + clearScreen():void

**Algorithm**: Print out a lot of new lines

**Method**: getCurvePoints

**UML Entry**: + getCurvePoints ():double

**Algorithm**: Create Scanner. Ask user how many points they would like applied as the curve. Return that number

**Method**: getCurvePercentage

**UML Entry**: + getCurvePercentage():double

**Algorithm**: Create Scanner. Ask user what percentage they would like applied as the curve. Return that number.

**Method**: percentageToPoints

**UML Entry**: + percentageToPoints (percentageInPoints:double, originalGrade:double):double

**Algorithm**: Set percentageInPoints equal to getCurvePercentage() \* originalGrade. Return percentageInPoints.

**Method**: applyCurve

**UML Entry**: + applyCurve (originalGrade:double, curveApplied:double):double

**Algorithm**: Return the sum of originalGrade and curveApplied.

**Method**: displayAdjustedGrade

**UML Entry**: + displayAdjustedGrade (curveApplied:double, adjustedGrade:double):double

**Algorithm**: Display the curveApplied and the adjustedGrade in a readable format