**Directions**: Launch BlueJ and load the project Z:\csa\csa\_exercises. *Create the project if it does not already exist.* We will create one class containing several arrays and methods.

Do not forget to test!!!

***Java Directions****: If you are doing this as a Java exercise, you will need to create a class file for each class and test it in a main method. Sample Java main method follows:*

*Public static void main(String[] args) {*

*}*

**Multidimensional Arrays**

**Create a 2d integer array** *(Syntax)*

Int[][] mArray = new int[*Number\_of\_Rows*][*Number\_of\_Columns*];

**Create a 2d integer array with 5 rows and 6 columns** *(Example)*

Int[][] mArray = new int[*5*][*6*];

**Access the value in the 3rd row and 4th column** *(Example)*

Int myValue = mArray[*2*][*3*];

**Set the value in the 4th row and 5th column** *(Example)*

mArray[*3*][*4*] = 90;

**Looping through a 2d array** *(Example)*

In order to loop through a 2d array, you must use nested looping (for or while)

for (… stepper1; arrayName.length; stepper1++) {

for (… stepper2; arrayName[0].length; stepper2++) {

*Most of the work is done here using the integer stepper variables from both loops.*

}

}

***Continued on next page***

**MultiArray Class**

In this exercise, you will create the MultiArray class. You will define several arrays and manipulate them in the methods you define.

1. **Define the Instance Variables** – They are as follows:
   1. **gradeArray –** an empty two dimensional integer array that can hold four rows and seven columns.
   2. **commentArray** - an empty two dimensional String array that can hold four rows and seven columns.
   3. Create an integer ***constant*** for four different students:
      1. CARISSA = 0
      2. TAYLOR = 1
      3. MOLLY = 2
      4. MICHAELA = 3
2. **Create the constructor method** – The constructor method will populate the arrays with default values. Use for nested loops to populate the following arrays as defined below:
   1. **gradeArray –** populate all cells of this array with random numbers between 20 and 100.
      1. **Note:** these numbers represent percentages earned on assignments.
   2. **commentArray** – populate this array with the text: “Comment: “ and add the row/column reference of the cell. The output should look like the following for row[1] column[4]:

Comment - R:1/C:4

**Note:** All loops in this exercise must *dynamically* work with arrays of different sizes (different numbers of elements). *You should have already learned to retrieve the length of the array in a previous exercise. Look it up if you must.*

1. **Create the printArrayGrid()** **methods** - These methods will be used to print the gradeArray *(integer)* and commentArray *(String)*, therefore it must be overloaded.*You should have already learned about method overloading in a previous exercise. Look it up if you must.*
   1. Create the version of the method that prints the values of a 2d integer array that is passed into it. Put at least two spaces between columns printed.
      1. ***Note****: The output should look like a grid of numbers.*
   2. Create the version of the method that prints the values of a 2d String array that is passed into it.
      1. ***Note****: The output should look like a grid of strings.*
   3. Nothing is returned from any of the above methods *(only printed)*.
   4. Test your methods.
2. **Create the printArrayGrids()** **method –** Create a method that calls the printArrayGrid() method one time for each array we are testing in this exercise.
   1. Check your output by calling printArrayGrids() at the end of your constructor. Does it look good???
3. **Create the updateArrays() method** – This method will modify our arrays. Code the following logic into this method.
   1. Modify gradeArray so that every student recieves 1en additional points for the first assignment.
      1. If a grade goes above 100, set it to 100. Test again here.
   2. Modify the last cell in the entire gradeArray grid to have the value 100. Make sure you do not use magic numbers. ***Use the length fields of the array to determine the last cell.*** Don’t forget to test.
   3. You will add calls to the **setComments()** and **printStudentDetail()** methods after we create them *(below)*
   4. **setComments()** method – Modify/update the commentArray so that the comments relate to the grade on the associated cell in the gradeArray:

|  |  |
| --- | --- |
| **Grade in gradeArray** | **Associated message in commentArray** |
| Above 89 | A - Very high quality work |
| 80 – 89 *(inclusive)* | B - Good work |
| 70 – 79 *(inclusive)* | C – You can do better |
| 60 – 69 *(inclusive)* | D - Please try harder |
| Below 60 | F - Stay after school to work with the teacher |

***Continued on Next Page***

* 1. **printStudentDetail()** method – Print the grade report for individual students. The report should look like the following
     1. The parameters to pass into the method include:
        1. The constant representing the student
        2. The name to use as a label when printing
     2. The report should look like the following:

Grade Report for: \*\* Carissa \*\*

99 A – Very high quality work

83 B – Good work

78 C – You can do better

…

Student Average: 92%

* + 1. Don’t forget to calculate and display the Student Average.
  1. Add the following code to the end of the updateArrays() method:
     1. setComments();
     2. printStudentDetail(CARISSA, "Carissa");
     3. printStudentDetail(TAYLOR, "Taylor");
     4. printStudentDetail(MOLLY, "Molly");
     5. printStudentDetail(MICHAELA, "Michaela");

1. **Testing**
   1. Create an instance of the ArrayTest class using the name arrayObj for the reference variable.
      1. ***Note****: There are two ways to test: Inside another method* ***or*** *within a main() method defined within this class.*
   2. Add the following lines of code after your instance has been created:
      1. arrayObj.printArrayGrids();
      2. arrayObj.updateArrays();
      3. arrayObj.printArrayGrids();
   3. Check the output to determine if everything worked.
      1. Were the grades increased by 10 for the first assignment?
         1. Did you repeat this test until you proved that no grades will go above 100?
      2. Do the comments work correctly?
      3. Check the output from printStudentDetail() carefully!