**Week 4 M5: Programming Assignment  
Large and Small Arrays**

Juan Macias Vasquez

Bellevue University

CSD402-H323 Java for Programmers (2261-DD)

**Jack Lusby**

September 6th, 2025

**Week 4 M5: Programming Assignment**

**Large and Small Arrays**

**GitHub Repository Link:**

<https://github.com/Juan551School/csd-402>

**Java Code**

//Juan Macias Vasquez

//Bellevue University

//CSD402-H323 Java for Programmers (2261-DD)

//Jack Lusby

//September 6th, 2025

**package** locatearray;

**public** **class** LandSarray{ // stands for Large and Small array couldn't fit it all

// Locate largest in double[][]

**public** **static** **int**[] locateLargest(**double**[][] arrayParam) {

**int**[] location = {0, 0};

**double** max = arrayParam[0][0];

**for** (**int** i = 0; i < arrayParam.length; i++) {

**for** (**int** j = 0; j < arrayParam[i].length; j++) {

**if** (arrayParam[i][j] > max) {

max = arrayParam[i][j];

location[0] = i;

location[1] = j;

}

}

}

**return** location;

}

// Locate largest in int[][]

**public** **static** **int**[] locateLargest(**int**[][] arrayParam) {

**int**[] location = {0, 0};

**int** max = arrayParam[0][0];

**for** (**int** i = 0; i < arrayParam.length; i++) {

**for** (**int** j = 0; j < arrayParam[i].length; j++) {

**if** (arrayParam[i][j] > max) {

max = arrayParam[i][j];

location[0] = i;

location[1] = j;

}

}

}

**return** location;

}

// Locate smallest in double[][]

**public** **static** **int**[] locateSmallest(**double**[][] arrayParam) {

**int**[] location = {0, 0};

**double** min = arrayParam[0][0];

**for** (**int** i = 0; i < arrayParam.length; i++) {

**for** (**int** j = 0; j < arrayParam[i].length; j++) {

**if** (arrayParam[i][j] < min) {

min = arrayParam[i][j];

location[0] = i;

location[1] = j;

}

}

}

**return** location;

}

// Locate smallest in int[][]

**public** **static** **int**[] locateSmallest(**int**[][] arrayParam) {

**int**[] location = {0, 0};

**int** min = arrayParam[0][0];

**for** (**int** i = 0; i < arrayParam.length; i++) {

**for** (**int** j = 0; j < arrayParam[i].length; j++) {

**if** (arrayParam[i][j] < min) {

min = arrayParam[i][j];

location[0] = i;

location[1] = j;

}

}

}

**return** location;

}

// To print int[][]

**public** **static** **void** printArray(**int**[][] array) {

**for** (**int**[] row : array) {

**for** (**int** value : row) {

System.***out***.printf("%4d", value);

}

System.***out***.println();

}

}

// To print double[][]

**public** **static** **void** printArray(**double**[][] array) {

**for** (**double**[] row : array) {

**for** (**double** value : row) {

System.***out***.printf("%6.2f", value);

}

System.***out***.println();

}

}

// Test program

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

**int**[][] intArray = {

{3, 5, 7},

{2, 9, 1},

{6, 4, 8}

};

**double**[][] doubleArray = {

{1.5, 2.7, 3.1},

{4.2, 0.9, 5.6},

{7.3, 2.8, 1.1}

};

// Print arrays first

System.***out***.println("Integer Array:");

*printArray*(intArray);

System.***out***.println();

System.***out***.println("Double Array:");

*printArray*(doubleArray);

System.***out***.println();

// Locate values

**int**[] largestIntLoc = *locateLargest*(intArray);

**int**[] smallestIntLoc = *locateSmallest*(intArray);

**int**[] largestDoubleLoc = *locateLargest*(doubleArray);

**int**[] smallestDoubleLoc = *locateSmallest*(doubleArray);

// Print results with values (I forgot the array starts at 0 so I kept redoing the project)

System.***out***.printf("Keep In Mind arrays start at 0 instead of 1 %n");

System.***out***.printf("Largest in int[][] = %d at (%d, %d)%n",

intArray[largestIntLoc[0]][largestIntLoc[1]],

largestIntLoc[0], largestIntLoc[1]);

System.***out***.printf("Smallest in int[][] = %d at (%d, %d)%n",

intArray[smallestIntLoc[0]][smallestIntLoc[1]],

smallestIntLoc[0], smallestIntLoc[1]);

System.***out***.printf("Largest in double[][] = %.2f at (%d, %d)%n",

doubleArray[largestDoubleLoc[0]][largestDoubleLoc[1]],

largestDoubleLoc[0], largestDoubleLoc[1]);

System.***out***.printf("Smallest in double[][] = %.2f at (%d, %d)%n",

doubleArray[smallestDoubleLoc[0]][smallestDoubleLoc[1]],

smallestDoubleLoc[0], smallestDoubleLoc[1]);

}

}

**Pictures of Code Running**

