**Northeastern Illinois University**

**CS-200: Programming I**

**Nested Loops and 2D Arrays**

**Problem 1.**

* Write a program that has the class name Problem1 and that has the main method. Leave

the main method empty for now.

* Write a method named noDuplicates that takes one parameter, an integer array a and

returns a new integer array.

* The method should create and return a new 1D array of integers with any duplicate values from a removed. Do not modify the input parameter a.
* Create a printArray method that takes a character array as a parameter and prints out the elements of the array on the same line separated by a space.
* Several sample usages are provided for you below. Use the sample usages in the main method to test your code (and use the printArray method to print out the results of  
  calling the noDuplicates method!).

|  |  |
| --- | --- |
| Sample method usage | Value of input array after method call |
| **int** [] b = {2, 2};  i**nt** [] b1 = noDuplicates(b); | {2} |
| **int** [] c = {1, 5, -4, 5, -4, 1, 0};  **int** [] c1 = noDuplicates(c); | {1, 5, -4, 0} |
| **int** [] d = { -2, 8, 9};  **int** [] d1 = noDuplicates(d); | { -2, 8, 9} |

**Problem 2.**

* Write a program that has the class name Problem2 and that has the main method. Leave

the main method empties for now.

* Write a method named transpose that takes one parameter, a 2-dimensional (2D) integer

array named a and returns a new 2D integer array.

* The method should create a new array a making the first column of a be the first row of

the new array, the second column of a be the second row of the new array, and so on. See

sample usage below.

* Create a printArray method that takes a 2D integer array as a parameter and prints out

the elements of each row on its own line separated by spaces.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the printArray method to print out the results of

calling the transpose method!).

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **int** [] [] a = {{3, -2, 18},  {14, 0, 9}};  **int** [] [] a1 = transpose(a); | {{3, 14},  { -2, 0},  {18, 9}}; |
| **int** [] [] b = {{5, 8},  {6, 6},  {0, 4},  { -1, -2}};  **int** [] [] b1 = transpose(b); | {{5, 6, 0, -1},  {8, 6, 4, -2} }; |

**Problem 3.**

* Write a program that has the class name Problem3 and that has the main method. Leave

the main method empty for now.

* Write a method named deepReverse that takes one parameter, a 2-dimensional (2D)

integer array named arr and returns a new 2D integer array.

* The method should create a new array a such that rows and columns are the reverse of

the array arr, such that first row of the array arr is the last row of the new array, second

row is the second last row of the new array and so on.

* Similarly, the first column in the array arr is the last column in the new array, second

column is the second last column in the new array and so on. See sample usage below.

* Create a printArray method that takes a 2D integer array as a parameter and prints out

the elements of each row on its own line separated by spaces.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the printArray method to print out the results of

calling the deepReverse method!).

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **int**[][] arr1 = {{1, 2, 4, 0},  {3, 4, 5, 6},  7, 8, 9, 12}};  **int**[][] a1 = deepReverse(arr1); | {{12, 9, 8, 7},  {6, 5, 4, 3},  {0, 4, 2, 1}}; |
| **int**[][] arr2 = {{2, 8},  {7, 20},  {9, 3},  {5, 12}};  **int**[][] a2 = deepReverse(arr2); | {{12, 5},  {3, 9},  {20, 7},  {8, 2}}; |

**Problem 5.**

* Write a program that has the class name Problem5 and that has the main method.
* Write a program that creates a 10 by 10 box. The box should look exactly as the sample

usage below.

* Even though the values are given, nothing should be hard coded - all constants must be

assigned a variable.

* Several sample usages are provided for you below.

|  |
| --- |
| 1 2 3 4 5 6 7 8 9 10  2 20  3 30  4 40  5 50  6 60  7 70  8 80  9 90  10 20 30 40 50 60 70 80 90 100 |