English name: ____________________

Unit 08: 2D Arrays – Worksheet 1

Unit 1: Primitive Types

To implement these questions correctly, you must consider type conversion.

1. Fill in the box to complete the code such that the method `printSum()` will correctly output the sum of the two numbers that are passed as parameters.

```
private static void printSum( int i, int j ) {
    System.out.println("The sum is: "   + (i + j)   );
}
```

2a. Fill in the box to complete the code such that the method `scorePercent()` will correctly return the student's score as a percent truncated to the nearest whole number. For example, if `score` is set to `20` and `max` is set to `30`, the method will return `66`.

```
private static int scorePercent( int score, int max ) {
    return   score * 100 / max   ;
}
```

2b. ***Challenge***: Fill in the box to complete the code such that the method `scorePercent()` will correctly return the student's score as a percent rounded to the nearest tenth. For example, if `score` is set to `20` and `max` is set to `30`, the method will return `66.7`. Use only type conversion, no method calls.

```
private static double scorePercent( int score, int max ) {
    return  (int)(score*1000.0 / max + 0.5) / 10.0  ;
}
```

Unit 2: Using Objects

3. Fill in the box to complete the code such that the method `printTwoLines()` will print to the console `line1` on the first line, followed by `line2` on the second line.

```
private static void printTwoLines( String line1, String line2 ) {
    System.out.println(   line1 + "\n" + line2   );
}
```

Unit 3: Boolean and `if` Statements

4. Fill in the box to complete the code such that the method `checkEquality()` will return `true` if the integer values of `i` and `j` are the same, and `false` if they are not the same. Note that this should work for all values, including large values, such as `999`.

```
private static Boolean checkEquality( Integer i, Integer j ) {
    return   i.equals(j);   ;
}
```

5. Fill in the box to complete the code such that the method `isAscending()` will return `"ascending"` if the integer values of `first`, `second` and `third` are increasing in value and `"not ascending"` if they are not.

```
private static String isAscending( int first, int second, int third ) {
    if (   first < second && second < third   ) {
    return "ascending";
} else {
    return "not ascending";
}
```

English name: ____________________

Unit 08: 2D Arrays – Worksheet 1

Unit 8: Two Dimensional Arrays

6. Using the method `Math.random()`, and no other methods, fill in the box to complete the code such that the method `randomInt()` will correctly return a random integer within the range [`min`, `max`].

```
private static int randomInt(int min, int max) {
    return  (int)(Math.random()*(max-min+1))+min  ;
}
```

7. Using enhanced for loops, fill in the box to complete the code such that the method `print2DIntArray()` will print out the two-dimensional integer array as a table with a space between numbers. For example, the array `a = { { 0, 1 } , { 2, 3 } }` will print as:

```
0 1
2 3
```

You can assume all numbers in the array are single digit so you do not need to take into consideration the number of digits in each number (which would mess up the alignment of columns if they differ).

```
private static void print2DIntArray(int[][] a) {
  for(int[] row: a) {
      for(int col: row ) {
          System.out.print(col + " ");
      }
      System.out.println();
  }
}
```

8. Using method `randomInt()` that operates as defined in question 1, fill in the box to complete the code such that method `random2DIntArray()` will generate and return a two-dimensional array populated with random values that are returned by successive calls to the method `randomInt()`. The returned array should have the number of rows and columns as specified in parameters `rows` and `columns`, respectively, and the random integers within the array should be within the range [`min`, `max`].

```
private static int[][] random2DIntArray(int rows,int columns,int min,int max) {
  int[][] a = new int[rows][columns];
  for(int row = 0; row < rows; row++) {
      for(int col=0; col < columns; col++) {
          a[row][col] = randomInt(min,max);
      }
  }
  return a;
}
```