Name ________________________________

CS 159 Advanced Programming
Spring '19
Midterm 1 Study Guide

1. True or False (2 point each)

____ The class diagram public visibility symbol is #.

____ An association line a UML class diagram can go from a class symbol to itself.

____ An array of `String` is a reference type while an array of `int` is a value type.

____ JUnit has assert methods for checking with a method returns true or false.

____ It is impossible to write a JUnit test that will determine whether code throws a particular exception.

____ Every try block must be accompanied by at least one catch block.

____ A try block can have a finally block and no catch block.

____ A catch block can be associated with two or more try bocks.

____ When a two dimensional array is declared, the number of columns may be specified along with the number of rows.

____ If `r` and `s` are regular expressions, then `(r|s)+` is also a regular expression.

2. Mark each of the following as a legal UML attribute specification (A), operation specification (O) or erroneous (E).

____ +x : StringBuilder[][]
____ p::name : String
____ Sin(double x) : double
____ +abs(v : double) : boolean, double
____ print(s : String)
____ ~v : Truck = null
____ x : int[1..5]
____ sum(a : int[*]) : int
____ p()
____ p() : void

3. Briefly explain the test driven development process.

4. Fill in the methods in the box below to illustrate the difference between the behaviors of reference and value (primitive) types in Java when variables of these types are passed as parameters to methods.

Name ______________________________

```
class X {

   private static void mValue(int y) {



   }

   private static void mReference(int[] b) {



   }

   public static void main(String[] ignored) {
      int x = 5;
      int[] a = {5};

     System.out.println("Initial value of x: "+x);
     System.out.println("Initial value of a[0]: "+a[0]);

      mValue(x);
      mReference(a);

     System.out.println("Final value of x: "+x);
     System.out.println("Final value of a[0]: "+a[0]);
   }
}
```

5. Make a UML diagram modeling as much of `Histogram` class from PA1 as you can.

6. Which of the following strings are matched by the regular expression `\w[0-9]\w*`?
   (2 points each)
   (a) 3.88
   (b) +3
   (c) A5xx
   (d) n0
   (e) STP
   (f) B2
   (g) F14
   (h) BB&T
   (i) 205

Name ______________________________

Consider the declarations below.

```
double percent;
char[] vowels = { 'a', 'e', 'i', 'o', 'u' };
boolean isPink;
String[] directions = { "north", "south", "east", "west" };
int[][] triangle = { {0}, {0, 1}, {0, 1, 2} };
```

7. For each of the expressions below, using the declarations above, indicate whether it is a value type (V), a reference type (R), or erroneous (E). (2 points each)

____ percent
____ triangle
____ isPink
____ directions[2]
____ vowels[5]
____ triangle[0]
____ triangle[][1]

8. **Sudoku** is game played on an *n*x*n* grid, wit*h n* > 0. A *solution* is a grid with the numbers from 1 to *n* listed once in each line and once in each column. A *puzzle* is a grid with some of the numbers of a solution filled in, but many spots in the grid left blank.

(a) Write a method `boolean isSudokuGrid(int[][] a)` that returns `true` if array `a` is a potential Sudoku grid. It must return `true` only when `a` is an array with the same number of rows and columns, and this number is at least 1. In all other cases (including when `a` is `null`), it must return `false`.

(b) Write a method `boolean isSudokuSolution(int[][] a)` that returns true if array `a` is a solution to a Sudoku puzzle. It must return `true` only if `a` is a Sudoku grid, and if `a` contains the numbers 1 through *n*, where n is the number of rows and columns in `a`, exactly once in each row and each column of `a`. In all other cases, it must return `false`. You may assume that you have a correctly written version of `isSudokuGrid()` available.

(c) Write a method void `printSudokuGrid(int[][] a, String fileName)` that attempts to write a Sudoku grid to the file named `fileName`, one row per line. If `a` is not a SudokuGrid, then the method must throw an `IllegalArgumentException`. If it cannot open the indicated file, or some error occurs while writing or closing the file, it must fail silently. If an element of `a` is less than 1 or greater than *n*, where *n* is the number of rows and columns in the grid, it must write a space for that element; otherwise, it must write the numeric value of the element. In all cases, it must write a space after each element of the grid, except possibly at the end of each line (though this is permitted).