

Name (PRINTED): _____

Student ID #: _____

Section # (or TA's:
name and time) _____

CMSC 131

Quiz #9

Fall 2009

Nov. 16

1. (12 points) Write a method which takes in a 2 dimensional array of doubles (`valList`), a single double (`val`) and a single integer (`place`). You may assume the `valList` is not null and that each of its rows is also non-null. You may also assume that `place` indicates a valid row in the `valList`. Your task is to insert the `val` into the row indicated by `place` if that value is not already present in that row.
 - If `val` is already present in the `place` row of `valList`, return a false without changing `valList`.
 - If `val` is not already present in the `place` row of `valList`, add `val` to the row indicated by the `place` parameter and return true.

```
public static boolean findName(double[][] valList, double val, int place){
```

```
}
```

2. (8 points) Assume you have an interface named "Hirable" and you are writing a class named "Employee" in that same package that you want to implement that interface.

Directions: Circle all of the statement below that are true (can be more than one) – put an X through the number of all statements below that are false (again, there can be more than one).

- (a) The Employee class must implement all methods and only the methods described in the Hirable interface.
- (b) The Employee class must implement all methods in the Hirable interface but can implement others.
- (c) The Hirable interface must contain a prototype for every method that is defined in the Employee class.
- (d) The Employee class must implement at least one of the methods that are in the Hirable interface, but only needs to implement the ones that are actually used in the driver.
- (e) A variable of Hirable type can reference a Hirable object on the heap.
- (f) A variable of Hirable type can reference a Employee object on the heap.
- (g) A variable of Employee type can reference a Employee object on the heap.
- (h) A variable of Employee type can reference a Hirable object on the heap.