Name: _____ Andrew Id: _____

15-121 Fall 2021 Quiz 5

Up to 25 minutes. Show your work. No calculators, no notes, no books, no computers, no other people.
Don't write import lines in free response problems.

1. (5 points) **Code Tracing**: Consider the following two classes, and indicate what is printed when the main method of the B class is executed. Place your answer (and nothing else) in the box under the code.

```
public class A {
    protected int a;
    protected int b;

    public A(int a, int b) {
        this.a = b;
        this.b = a;
        System.out.println("A: " + a + " " + b);
    }

    public int getOne() {
        return this.a;
    }

    public int getTwo() {
        return this.b;
    }
}

public class B extends A {
    public B(int c, int d) {
        super(d, c);
        System.out.println("B1: " + a + " " + b);
        a = d + 4;
        b = c - 4;
        System.out.println("B2: " + this.a + " " + this.b);
    }

    public int getOne() {
        return b;
    }

    public static void main(String[] args) {
        B bob = new B(15, 20);
        int e = bob.getOne();
        int f = bob.getTwo();
        System.out.println("e: " + e);
        System.out.println("f: " + f);
    }
}
```

2. **Free Response:** Consider the follow code designed to test two different classes: Vehicle and Motorcycle.

```
// You can create a vehicle with a name and capacity
Vehicle v = new Vehicle("Row Boat", 2);
System.out.println(v); // Prints: "Row Boat contains 0: "

// You can add passengers up to the capacity. The following two lines each add a
// passenger.
v.addPassenger("Ken"); // returns true
v.addPassenger("Barbie"); // returns true
System.out.println(v); // Prints: "Row Boat contains 2: Ken Barbie "

// Once the vehicle is full, you can't add any more passengers.
v.addPassenger("Ted"); // returns false
System.out.println(v); // Prints: "Row Boat contains 2: Ken Barbie "

// You can remove the most recently added passenger.
v.removePassenger();
System.out.println(v); // Prints: "Row Boat contains 1: Ken "

// If a passenger is removed, then space is freed up:
v.addPassenger("Ted");
System.out.println(v); // Prints: "Row Boat contains 2: Ken Ted "

// You can't remove from an empty vehicle
v.removePassenger(); // returns true
v.removePassenger(); // returns true
v.removePassenger(); // returns false
System.out.println(v); // Prints: "Row Boat contains 0: "

// A motorcycle is a vehicle that has an assumed name of Motorcycle and a
// capacity of 1
Motorcycle m = new Motorcycle();
System.out.println(m instanceof Vehicle); // Prints: "true"
m.addPassenger("Mo"); // returns true
m.addPassenger("Fa"); // returns false
System.out.println(m); // Prints: Motorcycle contains 1: Mo

// While a motorcycle is doing a wheelie, you can't remove a passenger
m.popWheelie(); // No value returned here
m.removePassenger(); // returns false

// Once the wheelie stops, you can remove as usual for vehicles
m.stopWheelie(); // No value returned here
m.removePassenger(); // returns true
m.removePassenger(); // returns false
```

Write the classes Vehicle and Motorcycle so that the test code runs as specified. Do not hardcode against the values used in the testcases, though you can assume the testcases cover the needed functionality. For full credit, you must use proper object-oriented design, including good inheritance and avoiding unnecessary code duplication.

Note that it is possible to receive full points for the Motorcycle class, even if your Vehicle class is completely wrong.

Write your answers in the appropriate locations on the next page.

(a) (8 points) The Vehicle Class

(b) (7 points) The Motorcycle Class