Quiz 6 – CS 315

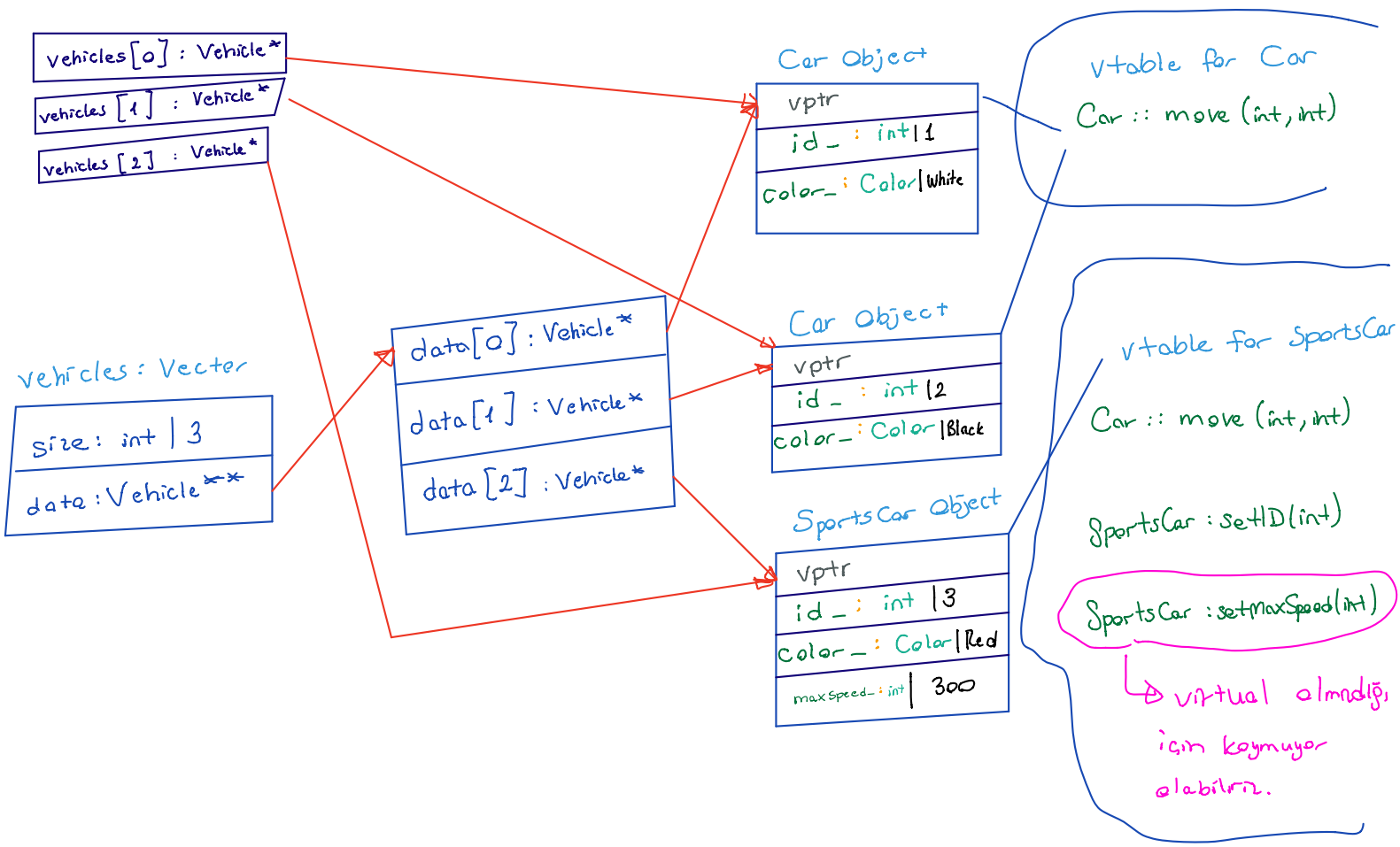
Name: Gas, ID: 911, Section: Porshe

Consider the following code:

```
enum class Color { White, Black, Red };
class Vehicle {
protected:
    int id_;
public:
    Vehicle(int id) { ... }
    virtual void setId(int id) { ... }
    virtual void move(int dx, int dy) = 0;
};
class Car : public Vehicle {
protected:
    Color color_;
public:
    Car(int id, Color color) { ... }
    void setColor(Color color) { ... }
    void move(int dx, int dy) { ... }
};
class SportsCar : public Car {
protected:
    int maxSpeed_;
public:
    SportsCar(int id, Color color,
              int maxSpeed) { ... }
    void setMaxSpeed(int maxSpeed) { ... }
    void setId(int id) { ... }
};
```

```
void main()
{
    using namespace std;

    vector<Vehicle*> vehicles(3);
    vehicles[0] = new Car(1, Color::White);
    vehicles[1] = new Car(2, Color::Black);
    vehicles[2] = new SportsCar(3, Color::Red, 300);

    // location 1

    for (Vehicle * vehicle : vehicles)
        vehicle->move(3, -4); // location 2

    vehicles[0]->setId(4); // location 3
    vehicles[1]->setId(5); // location 4
    vehicles[2]->setId(6); // location 5

    for (Vehicle * vehicle : vehicles)
        delete vehicle;
}
```

a) Draw the memory contents at location 1, showing the virtual table pointers and the virtual tables. Assume that vector class has two members: a data (which is an array) and a size (which is an int).



b) Explain how the call to move at location 2 is generated (compile-time) and resolved (run-time).

The idea is to generate an indirect call.:

- As part of each object, a *virtual table pointer* is kept.
- The virtual table pointer points to a table of function descriptors (aka the *virtual table*)
- The call is made by following the virtual table pointer, finding the appropriate function from the virtual table, and making a call to it.
- There is one virtual table per non-abstract class (not one per object).

- _vptr is a hidden pointer added by the compiler to the base class. This pointer points to the virtual table of that particular class.
- each object of a class with virtual functions transparently stores this _vptr.
- call to a virtual function by an object is resolved by following this hidden _vptr

c) Which `setId`'s are called at locations 3, 4, and 5?

if the function is declared as virtual, and the children class implements the method; then the children's is called.

if the function is implemented in the children but the reference which holds the children object is the parent;

```
Animal *cat = new Cat;
cat->eat(); // would call Animal's implementation if
```

eat() is not declared as virtual in Animal.

Sadede Gel;

3, 4 → Vehicle

5 → SportsCar