

Name: \_\_\_\_\_, ID: \_\_\_\_\_, Section: \_\_\_\_\_

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).

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