

Advanced Object Oriented Programming – Inheritance, Polymorphism, Override vs. Overload, instanceof

1. Study the following classes (all of which compile correctly) and then answer the questions that follow them.

```
class Vehicle {
    private int numWheels;

    public Vehicle (int nw) {
        numWheels = nw;
    }

    public int getNumWheels() {
        return numWheels;
    }

    public boolean equals(Vehicle other) {
        return other != null &&
            numWheels == other.numWheels;
    }

    public boolean isBig() {
        return numWheels > 4;
    }
}

class Truck extends Vehicle {
    private int loadCapacity;

    public Truck (int nw, int lc) {
        super(nw);
        loadCapacity = lc;
    }

    public int getLoadCapacity() {
        return loadCapacity;
    }

    public boolean equals(Truck other) {
        return other != null &&
            loadCapacity == other.loadCapacity;
    }
}
```

```

class Car extends Vehicle {
    private num numSeats;

    public Car (int ns) {
        super(4);
        numSeats = ns;
    }

    public int getNumSeats() {
        return numSeats;
    }

    public boolean isBig() {
        return numSeats > 5;
    }
}

```

Consider each of the following fragments to be in a `main` method that uses these classes. All the constructors are valid. The other statements fall into one of the following categories.

- i) The statement will compile and execute correctly.
- ii) The statement will not compile but it can be repaired by a cast.
- iii) The statement will not compile and cannot be repaired by a cast.
- iv) The statement will compile but fails to execute correctly.

For each fragment, identify the category to which the second statement belongs.

- a) `Vehicle v1 = new Vehicle(4);`
`int nw = v1.getNumWheels();`
- b) `Car c1 = new Car(6);`
`System.out.println(c1.getNumWheels());`
- c) `Vehicle v2 = new Car(4);`
`int capacity = v2.getNumSeats();`
- d) `Truck t1 = new Truck(18, 14000);`
`int loadLimit = ((Vehicle)t1).getNumSeats();`
- e) `Car c2 = new Car(5);`
`int maxLoad = (((Truck)c2).getLoadCapacity());`
- f) `Vehicle v3 = new Vehicle(4);`
`System.out.println(((Car)v3).getNumSeats());`
- g) `Vehicle v4 = new Truck(4, 1200);`
`System.out.println(((Truck)v4).getNumWheels());`
- h) `Truck t2 = new Truck(10, 8000);`
`Car c3 = (Car)t2;`

Assume the following fragment has been run,

```
Vehicle v1 = new Vehicle(4);  
Vehicle v2 = new Vehicle(4);  
Vehicle v3 = new Vehicle(6);  
Vehicle v4 = new Truck(10, 500);  
Truck t1 = new Truck (10, 1000);  
Truck t2 = new Truck (10, 700);  
Vehicle v5 = new Car(8);  
Car c1 = new Car(8);  
Car c2 = new Car(6);
```

What value does each of the following method call return? Or indicate if the method call causes an error.

- i) `v1.isBig();`
- j) `v4.isBig();`
- k) `t1.isBig();`
- l) `v5.isBig();`
- m) `c1.isBig();`
- n) `v1.equals(v2);`
- o) `v1.equals(v3);`
- p) `v1.equals(t1);`
- q) `v1.equals(c1);`
- r) `v3.equals(c1);`
- s) `v4.equals(v2);`
- t) `v4.equals(t1);`
- u) `t1.equals(v3);`
- v) `t1.equals(v4);`
- w) `t1.equals(t2);`
- x) `t1.equals(c1);`
- y) `c1.equals(v1);`
- z) `c1.equals(t1);`