

## Practice: Understanding Interfaces – Solutions

Refer to the problem handout for the code that these questions are asking about.

1. **Question:** What methods do you need to add to each of `Car` and `Bicycle` to get this code fragment to compile (setting aside the missing constructors)?

**Answer:** The two methods listed in the interface (`tuneUpCost` and `canCarry`).

2. **Question:** Does having a class implement an interface change how you write its constructor?

**Answer:** No. Interfaces affect what method you must provide in a class, but they do not impact the constructors.

3. **Question:** Should `builtBefore` be added to the `IVehicle` interface? Why or why not?

**Answer:** No. Only methods that make sense for all classes that implement the interface should be included in the interface. `builtBefore` is not meaningful for bicycles (since they have no information about when they were built).

4. **Question:** In the `VehicleTest` class, you want to define a `Bicycle` as follows:

```
_____ newKidsBike = new Bicycle(0, 1);
```

Which types can you use in the blank line (without causing a compile error just with this single line)?

**Answer:** Either `Bicycle` or `IVehicle`.

5. **Question:** Assume you defined a `Car` object as follows:

```
IVehicle oldCar = new Car(200000, 1995, 2);
```

Which methods can you call on `oldCar`, given the type that you gave it?

**Answer:** Since `oldCar` has type `IVehicle`, you can only call methods that are listed in `IVehicle`. So you can call `tuneUpCost` and `canCarry`, but not `builtBefore`, even though the `builtBefore` method is within the `oldCar` object.

6. **Question:** Assume you defined a `Car` object as follows:

```
Car oldCar = new Car(200000, 1995, 2);
```

Which methods can you call on `oldCar`, given the type that you gave it?

**Answer:** Since `oldCar` has type `Car`, you can only call methods that are defined in the `Car` class, which must include those listed in `IVehicle`. You can call each of `tuneUpCost`, `canCarry`, and `builtBefore`,

---

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS18 document by filling out the anonymous feedback form: <https://cs.brown.edu/courses/cs018/feedback>.