

## Lab 1

### Exercise 1.2

#### Q1

We used an Abstract Class because all the vehicles share common attributes and methods.

The shared attributes:

- GPM
- Tank\_Size
- Gas
- Make
- Color

The shared methods:

- fillGas
- drive
- changeTires

An interface would not let us define the common methods.

#### Q3

The error you're encountering is a `ClassCastException`. This occurs because you are trying to cast an instance of `Truck` to a `Car`, which is not valid since `Truck` and `Car` are siblings in the inheritance hierarchy and not directly related.

To avoid such problems, you can use the `instanceof` operator to check the type of the object before casting it. This is a runtime check that ensures the object is of the expected type before performing the cast.

Here's how you can modify your code to include the `instanceof` check:

```
Vehicle v = new Truck();
if (v instanceof Car) {
    Car c = (Car)v;
    // Safe to use c as a Car
} else {
    System.out.println("The object is not of type Car.");
}
```

By adding this check, you ensure that the cast only occurs if `v` is an instance of `Car`, avoiding the `ClassCastException`.

Itay Shamir 213147481  
Sivan Zusin 318934312

## Reasons to use SybType:

### Encapsulation

Subtypes help encapsulate behavior specific to a certain type, making the code more modular and easier to understand. Each subclass can have its own unique methods and properties while sharing common functionality defined in the abstract class.

### Maintainability

Using subtypes makes the code easier to maintain. Changes in the common behavior can be made in the abstract class, and these changes will automatically propagate to all subclasses. This reduces the risk of inconsistencies and errors.

### Avoiding ClassCastException

By using subtypes correctly and leveraging polymorphism, you can avoid runtime errors like `ClassCastException`. Instead of casting, you can use the methods defined in the abstract class or interface, ensuring that the correct implementation is called for the actual object type.

## Exercise 2.2

