**Question 1:** What is the difference between an abstract class and an interface?

**Question 2: i).** Give an account of advantages and disadvantages of inheritance and composition. **ii).**

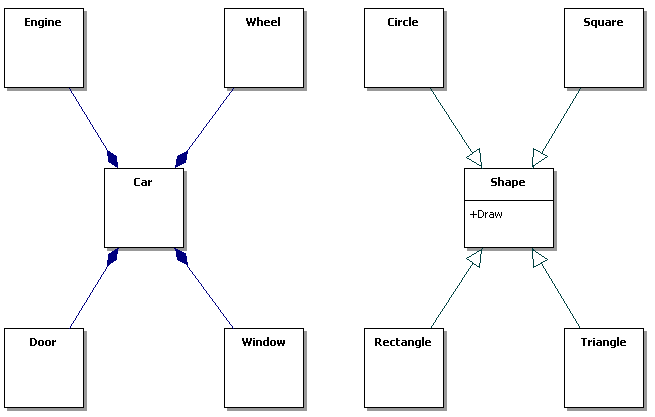
Give an example of when inheritance is better and when composition is better

**Advantages:**

|  |  |
| --- | --- |
| Composition | Inheritance |
| Easier to change the interface of back-end classes and front-end classes. | Dynamic binding and polymorphism can be used, and makes the code easier to change if the needed change involves adding a new subclass |
| Allow us to delay the creation of back end objects until they are needed | Easy to build and describe Is-A relationship |

**Disadvantages:**

|  |  |
| --- | --- |
| Composition | Inheritance |
| Cannot use dynamic binding and polymorphism | Super class’s interface changing will cause many modifications |
|  | Encapsulation is fragile |



On the left, we want to describe a car system with engine, wheels, doors, etc. We can use composition rather than inheritance to explain and describe the relationship. On the right, it’s better using the inheritance relation to build and implement the system because it shows several obvious “Is-A” relationships and we can easily using the polymorphism here.