

# Composition

---

Instead of using private inheritance, a better solution is to use composition. **Composition creates a new class by combining simpler classes**, using instances of the simpler class as the data members. In composition, an object is **composed of other objects**, which make up its "parts." That's why it's called (informally) a **has-a** relationship; because we can say that:

- A car **has a** motor, or
- A bicycle **has a** seat, or
- A computer **has a** CPU



Here is some code for a version of the **GiPod** which uses composition; note that it needs to **explicitly write the prototypes** for any member functions that it uses.

```
class GiPod
{
public:
    void playMusic() const
    {
        caddy.playMusic(); // forward or delegate
    }
private:
    Cadillac caddy;        // data member
};
```



This course content is offered under a CC Attribution Non-Commercial license. Content in this course can be considered under this license unless otherwise noted.