**virtual** **bool** collisionAvoidanceWorthy() { **return** **true**; }

* I put this function in the Actor base class because all Actors have property of either being collisionAvoidanceWorthy or not
* I made this function virtual because most Actors have this property, but some Actors do not (in which case, they can override this function)

**virtual** ~Actor() {}

* I put this function in the Actor base class because all Actors have a destructor
* The Actor destructor must be virtual so that the subclasses can also have their own destructors

**void** setToDead() { m\_isAlive = **false**; }

* I put this function in the Actor base class because all actors have the property of alive-ness
* I made this function non-virtual because its implementation is the same for all Actors

**double** getVertSpeed() { **return** m\_vertSpeed; }

* I put this function in the Actor base class because all actors have the property of vertical speed
* I made this function non-virtual because its implementation is the same for all Actors

**virtual** **void** receiveDamage(**int** damage) {}

* I put this function in the Actor base class because
* I made this function non-virtual because its implementation is the same for all Actors

**virtual** **void** doSomething() = 0;

**bool** isAlive() { **return** m\_isAlive; }

**int** getLaneNum();

**virtual** **bool** isAffectedProjectiles() = 0;

**bool** doOverlap(Actor\* otherActor);