

```
{
    Step::RefPtr<Entity> entity = new Entity(); // allocation on the heap
    std::cout << entity.referenceCount() << std::endl; // prints 1 ref
    Step::RefPtr<Entity> same_entity = entity.get();
    std::cout << entity.referenceCount() << std::endl; // prints 2 ref
    same_entity = NULL;
    std::cout << entity.referenceCount() << std::endl; // prints 1 ref
}
// entity gets out of scope and the underlying pointer to the Entity gets
// deleted because if reference count falls to 0.
```