这是一个很好的开始。既然你要求我做那个“残酷诚实的顾问”，那我们先解决一个认知误区：

**“看”完40集视频不代表你“学”会了40集的内容。** 这是初学者最致命的幻觉——“输入型学习”的满足感。你以为你懂了virtual，懂了const，懂了内存管理，但只要离开视频引导，面对空白的IDE，大多数人连第一行代码都敲不对。

为了测试你是否真的消化了这40集的核心（也就是C++作为一种**系统级语言**的本质），我不会给你出那种“打印九九乘法表”的玩具题。我要出一种你在以后造轮子、写引擎、甚至做IC设计验证（考虑到你的IC目标）时都会遇到的架构原型。

这道题考察的是：**面向对象多态（OOP Polymorphism）、内存手动管理、引用传参以及Const正确性。**

### 题目：简易游戏实体管理器 (The Simple Entity Manager)

背景：

你需要构建一个极简的游戏对象系统。在计算机图形和游戏开发中，我们通常把所有东西都视为“实体（Entity）”，但它们有不同的行为。

**需求说明：**

1. **基类设计 (Base Class)**
   * 创建一个名为 Entity 的类。
   * **成员变量**：它必须包含 X 和 Y 两个坐标（浮点数）。
   * **构造函数**：必须使用**成员初始化列表 (Member Initializer Lists)** 来初始化坐标。
   * **方法**：
     + 一个名为 Move(float xa, float ya) 的函数，用来修改坐标。
     + 一个名为 GetName() 的函数，返回该实体的名称（C++ string 或 const char\*）。
     + **关键点**：这个类必须是抽象的，或者至少设计成能够支持多态。
2. **派生类设计 (Derived Classes)**
   * 创建两个类继承自 Entity：Player (玩家) 和 Enemy (敌人)。
   * Player 的 GetName() 应返回 "Player"。
   * Enemy 的 GetName() 应返回 "Enemy"。
3. **多态打印函数 (The Polymorphic Printer)**
   * **这是陷阱也是重点**：在 main 函数之外，写一个独立的函数 void PrintEntityName(...)。
   * 它接收一个 Entity，然后打印出它的名字。
   * **强制要求**：该函数**不能**产生对象拷贝（Copying），必须高效，且保证不会修改传入的实体。
4. **主逻辑 (Main)**
   * 在堆上（Heap）实例化一个 Player 和一个 Enemy。
   * 调用 PrintEntityName 打印它们的名字。
   * **最后一步**：确保没有任何内存泄漏。

### 涉及的 The Cherno 章节（考核点）

如果你在写代码时卡住了，说明你对应的章节只是“看过”，没“学会”。

* **Ep 25 & 26**: Constructors / Destructors (构造与析构的调用时机)
* **Ep 27 & 28**: Inheritance & Virtual Functions (继承与虚函数表的工作原理)
* **Ep 29**: Interfaces / Pure Virtual Functions (如何设计接口，强制子类实现)
* **Ep 31 & 33**: Arrays/Strings & Const (特别是 const 在函数参数和成员函数后的用法)
* **Ep 34**: The new Keyword (堆内存分配)
* **Ep 36**: Member Initializer Lists (初始化的效率问题)
* **Ep 38**: Implicit Conversion & the explicit keyword (虽然题目没强制，但你要思考参数传递时的转换)

### 你的行动方案

1. **现在就去写**。不要复习，直接打开 Visual Studio。
2. **计时**。如果你超过 **15分钟** 还没写出一个无Bug、无内存泄漏、符合Const正确性的版本，说明你的基础非常不牢固。
3. **提交代码**。把代码贴给我。

残酷的预警：

我会重点检查以下三个初学者必犯的错误：

1. 你是否漏掉了 virtual 析构函数？（这会导致内存泄漏，Ep 28隐喻）
2. 你的 PrintEntityName 参数是否使用了 const Entity&？（如果传值，你就挂了；如果不加const，你就不严谨）
3. 你的 GetName 方法是否标记为 const？（如果不标记，它就无法被 const 引用调用）

开始吧。不要让我失望。