1. 定义一个时间类 **Time**

**数据成员（访问权限定义为protected）**：

时、分、秒

**成员函数（访问权限定义为public）：**

* **带默认参数的构造函数**：数据成员初始化

**Time**(double \_hour = 0.0, double \_minute = 0.0, double \_second = 0.0);

* **拷贝构造函数**：数据成员初始化 **Time**(const **Time** &time);
* 设置小时
* 设置分钟
* 设置秒
* 获取小时（定义为const成员函数）
* 获取分钟（定义为const成员函数）
* 获取秒（定义为const成员函数）
* 打印时间信息（定义为const成员函数），**格式**：**09:02:03**

**要求**：将类的定义与类成员函数的实现分开。

1. 定义一个图书类**Book**

**数据成员（访问权限定义为protected）**：

书名（char \*bookname）、价格、图书数量（**size\_t num**）、图书馆总存书数量（**static size\_t total**）

**成员函数（访问权限定义为public）**：

* **构造函数**：对书名、价格和图书数量进行初始化

**Book**(char \*\_bookname, double \_price, size\_t \_num);

**注意**：对静态数据成员如何处理？

* **拷贝构造函数**：数据成员初始化（避免**浅拷贝**）

**Book**(const **Book** &book); **注意**：对静态数据成员如何处理？

* **析构函数**：释放系统资源

**注意**：对静态数据成员如何处理？

* 借书操作 void borrow(size\_t bnum); **注意**：对静态数据成员如何处理？
* 还书操作 void restore(size\_t bnum); **注意**：对静态数据成员如何处理？
* 获取图书价格（定义为const成员函数）
* 获取图书数量（定义为const成员函数）
* 获取图书馆总存书数量（定义为静态成员函数）
* 显示图书信息（定义为const成员函数）

**要求**：将类的定义与类成员函数的实现分开。

1. 定义一个二维坐标点类**Point**

**数据成员（访问权限定义为protected）**：

横坐标、纵坐标

**成员函数（访问权限定义为public）**：

* + **带默认参数的构造函数**：数据成员初始化

**Point**(double \_x = 0.0, double \_y = 0.0);

* + **拷贝构造函数**：数据成员初始化 **Point**(const **Point** &point);
  + 设置横坐标
  + 设置纵坐标
  + 获取横坐标（定义为const成员函数）
  + 获取纵坐标（定义为const成员函数）
  + 打印点的坐标信息（定义为const成员函数），格式：(x, y)

**要求**：将类的定义与类成员函数的实现分开。

定义一个普通函数用来计算两个**Point**对象的距离:

double pdistance(const **Point** &p1, const **Point** &p2);

并将pdistance函数声明为类**Point**的友元函数，使其能够直接访问**Point**的保护数据成员。

定义一个普通函数用来计算两个**Point**对象的中点：

**Point** midpoint(const **Point** &p1, const **Point** &p2);

并将midpoint函数声明为类**Point**的友元函数，使其能够直接访问**Point**的保护数据成员。