手写答案测试自己 - 笔

*你的名字*_____

全日制学习　　*非全日制学习*　　***(删除一项)***

所有问题均为**手写**答案。需要 C++ 编码。理论/概念题需要用 C++ 代码演示理论/概念。

先**手写**答案，包括代码。使用精心手写的测试计划对手写代码进行桌面检查。不要先在电脑上键入代码，因为你可能会产生一种错觉，以为自己已经学会了，其实可能并没有。有一些关于手写与键盘的研究。请参阅 Mangen 等人的研究；《纽约时报》；Kiefer 等人的研究；《卫报》也报道了大学生测试对象的研究；Mackenzie 的研究；发展认知技能的手写；与打字相比的优势；脑电图研究。

重要建议

- **在编写代码以解决给定问题或编写数据结构代码时，还可能要求您提供测试计划、测试程序以及数据结构或类的模块化单元测试。测试计划/程序应检查正常运行情况，还应中断任何无效模块或被测数据结构。"中断 "也可以指异常终止。测试计划和程序都是手写的。**
- **仅仅知道理论/概念几乎没有分数分配。大部分分数都用于使用 C++ 代码演示理论/概念，以及使用代码进行解释--解释代码如何与理论/概念相匹配。**
- **仅仅代码正确还不足以通过考试。应用理论概念的正确设计至关重要。举个最糟糕的例子，如果把所有内容都写在一个 *main()* 函数中，即使程序运行完美，分数也**

会是 0 分。如果使用全局变量，也会得到同样的结果。

- **按要求回答问题。如果背诵的代码与问题无关，则不会得分。认为自己在考试中写了很多代码就能通过考试是一个常见的错误。因此，请仔细阅读问题，了解问题的要求，然后根据要求撰写答案。你写的答案越是错误，我们就越相信你不知道答案。**
- **任何在类声明正文中包含实现的编码答案，分数都将减半。只允许使用方法原型。与 Java 不同，实现在类声明之外。如果您认为需要内联任何方法，请在答案中提供书面说明。**

- **考试时，可以用铅笔写出大致答案。希望做标记的答案应使用黑色或蓝色钢笔书写。**

1. 编写一个 C++ 程序，提示用户以时、分、秒为单位输入某一事件的经过时间。然后输出以秒为单位的经过时间。本题**不**需要面向对象的解决方案。您将使用哪些测试数据来确保您的程序正常运行。您需要提供选择测试数据的理由。

2. 编写一个面向对象的 C++ 程序，提示用户以秒为单位输入某一事件的经过时间。然后程序会输出以小时、分钟和秒为单位的经过时间。您将使用哪些测试数据来确保您的程序正常运行。您需要说明选择测试数据的原因。

3. 编写一个 OO C++ 程序，从名为 cents.txt 的数据文件中读取整数列表。数据文件中的数字用空格或换行符分隔。数据文件中的每个数值都代表以分为单位的金额。

   您的程序会打印出：以美分为单位的数值，以及该数值所代表的美元和美分。

   当文件 cents.txt 中有两个数字时，程序的输出示例，但在其他情况下可能会有更多数字。
   10 235

   输出：
   10 美分等于 0 美元 10 美分
   235 美分等于 2 美元 35 美分

   描述用于解决上述问题的数据结构。可以使用哪些其他数据结构，你选择的数据结构与其他数据结构相比有哪些优势？

   您的解决方案中可以使用什么设计模式？这种模式如何用于可能涉及不同转换的其他解决方案？使用 UML 图表进行解释。

   解释您的解决方案是如何设计以适应模型-视图-控制器 (**MVC**) 模式的。您的解释需要明确指出哪些类、子程序（函数、过程）适合 MVC 的哪个部分。您还应解释为什么它们适合 MVC 组件。

为应用程序编写测试计划。

4. 一个足球场的经理希望你编写一个程序，计算每场比赛后的门票销售总额。门票有四种类型，每种类型都有自己的价格。每场比赛结束后，数据会存储在名为 *sales.txt 的*数据文件中。数据文件格式如下所示：

票价 售出票数

.....

示例数据如下。您不能认为这就是每场比赛后的数据。只有数据，没有列标题。同一行的数据项之间用空格隔开。

```
250   5750
100   28000
50    3570
25    18750
```

第一行显示票价为 250 美元，共售出 5750 张。您的程序将输出售出的门票数量和总销售额。

解释您的解决方案是如何设计以适应模型-视图-控制器 (**MVC**) 模式的。您的解释需要明确指出哪些类、子程序（函数、过程）适合 MVC 的哪个部分。您还应解释为什么它们适合 MVC 组件。

5. 使用 UML，设计一个生日列表来记录任何人（朋友、家人、同事等）或宠物的生日。在开始设计之前，请首先考虑生日列表的内容。

   想一想所使用的类、结构和数据结构，以及要存储哪些信息。你的程序有哪些菜单选项?

   应尽可能使用最高级别的抽象。使用 UML 确定不同的类及其关系。

   用 C++ 实现你的生日列表，并编写一个测试程序来证明你的生日列表是有效的。使用代码注释说明您的解决方案的哪些部分属于 **MVC** 的哪些组件。

6. UML 允许我们在面向对象设计中描述类的关系。请使用 UML 解释两种此类关系：*专业化*和*实现*。您必须绘制 UML 图。

   a. 在什么情况下您会使用 "*专业化*"?
   b. *实现* "与 "*专业化*"有何不同?

7. **使用 C++** 代码，实现上一问题中下列关系中的每一种关系的示例：*专业化*; *实现* - *实现* UML 设计。

8. 使用您选择的任何相关 C++ 代码示例（或多个示例），解释 "以最小但完整

的方式设计或编码类 "的含义。使用适当的 C++ 代码示例（或多个示例），解释如何在不创建额外方法或类的好友的情况下为类提供额外功能?

您将如何使用命名空间来管理上述设计?

9. 使用 C++ 示例，解释 C++ 模板如何提供抽象机制。

10. 解释为什么 C++ 等语言提供模板机制。提供一个 C++ 模板子程序（不得是类的方法）的示例，以演示模板的效用。您需要提供解释和代码。

11. 讨论 "类是一种数据类型 "这一说法的有效性。用 C++ 示例来说明你的答案。仅仅编写代码是不够的。您需要解释该语句有效或无效的原因。

12. 使用 C++ 编写一个完整的**最小**但**完整的**模板向量类。就我们而言，模板矢量类是封装在矢量类中的动态数组。Vector 类需要最小但完整。公共方法不应提供重叠或多余的功能。你可以决定并维护哪些是必要的。使用 C++，为 Vector 类提供一个完整的模块化单元测试。**不得**使用 STL 数据结构。

13. 使用 C++ 编写一个完整的向量类模板。向量类需要简约但完整。公共方法不应提供重叠或多余的功能。使用 C++ 为 Vector 类提供一个完整的模块化单元测试。**必须**在 Vector 类中**使用**适当的 STL 数据结构。由你决定并为必要的数据结构辩护。解释为什么这种数据结构是合适的。

14. 用 C++ 编写一个使用 Vector 类的应用程序。该程序将首先读取存储在名为 *titles.txt 的*数据文件中的所有书名。这些数据存储在 Vector 中。然后，应用程序会按照 Vector 中的存储顺序将书名打印到屏幕上。数据文件的每一行都有一个书名。数据文件中可以有任意数量的标题。数据文件中的数据示例如下：

    设计模式
    C++ 算法与数据结构 神经计算高级方法

    解释您的解决方案是如何设计以适应模型-视图-控制器 (**MVC**) 模式的。您的解释需要明确指出哪些类、子程序（函数、过程）适合 MVC 的哪个部分。您还应解释为什么它们适合 MVC 组件。

15. 编写一个名为 *fibonacci 的*C++ 子程序，以 n 为参数，用 STL 向量返回斐波那契数列中的前 n 个数字。因此，fibonacci(7) 将在 STL 向量中返回 0、1、1、2、3、5、8。编写一个测试程序来测试你的例程。您**不**需要设计一个 OO 解决方案。唯一需要的对象就是

STL 向量对象。

16. 使用 C++ 编写一个最小但完整的二进制搜索树（BST）模板。使用 C++ 为 BST 编写一个完整的模块化单元测试。其中一个测试需要一个日期 BST。因此，您还需要编写一个日期类。使用 BST<Date> 测试您的 BST。

17. 为 BST 实现递归插入、删除树和遍历。

18. BST 搜索方法的运行时性能（Big-O）是多少？什么情况会影响这种运行时性能，从而产生最坏情况下的 Big-O？最坏情况下的 Big-O 性能是多少？

19. 为执行以下 BST 操作的方法提供递归 C++ 实现：顺序内、顺序后和顺序前遍历；插入到树中和删除整棵树。

20. 使用 UML，设计一个**最小**但**完整的**链表数据结构。用 C++ 实现模板链表。

21. 使用 C++ 示例解释**继承**的概念。您的代码应演示继承是如何工作的。

22. 使用 C++ 示例解释**动态**多态性的概念。您的代码应演示多态是如何工作的。使用代码注释解释动态多态性发生的关键要求。

23. 用 C++ 示例解释堆栈和队列的本质区别。堆栈能否使用队列来实现，反之亦然？用 C++ 代码演示如何实现。

24. 使用 UML 符号，为堆栈数据结构设计一个最小但完整的模板类。类设计必须显示数据结构的所有细节。所有公共方法都应适当命名。必须显示参数和返回类型。Stack 类必须包含的两个方法是 "push "和 "pop"。

25. 使用 UML 符号，为队列数据结构设计一个最小且完整的模板类。类设计必须显示数据结构的所有细节。所有公共方法都应适当命名。必须显示参数和返回类型。队列类必须包含的两个方法是用于加入和离开队列的 "join "和 "leave"。

26. 使用 C++ 编写一个**最小**但**完整的**模板堆栈类和一个**最小**但**完整的**模板队列类。**不要**使用任何 STL 数据结构。你可以决定并维护哪些是必要的。为每个类提供单独的模块化单元测试。**单元测试程序应检查功能是否正常，并能中断**设计或实现不正确的数据结构。"中断 "也可以指异常终止。

27. 使用 C++ 编写一个**最小**但**完整的**模板堆栈类和模板队列类。您**必须**在每个类中**使用**最合适的 STL 数据结构。你决定并维护哪些是必要的。解释哪些 STL 数据结构最适合您的每个类。为每个类提供单独的模块化单元测试。**单元测试程序应检查功能是否正常，并能中断**设计或实现不正确的数据结构。"中断 "也可以指异常终止。

28. 解释内聚和耦合的概念。就这些概念而言，软件设计师的目标是什么？用你自己的

C++ 示例来解释需要实现的目标。

29. 什么是内存泄漏？使用 C++ 示例演示内存泄漏。

30. 什么是悬空指针？使用 C++ 演示如何出现悬空指针。

31. 使用 C++ 示例解释深拷贝和浅拷贝的概念。

32. 如果需要编写析构函数，还需要编写哪些方法或操作符？用 C++ 代码解释你的答案。为什么需要析构函数？

33. 如果一个类有指针数据，该类必须提供哪些方法（说出它们的名称）？用你自己的 C++ 示例解释为什么需要这些方法，以及这些方法的作用。如果不提供这些方法会发生什么？用代码注释来解释。

34. 在 OO 设计中，SOLID（S.O.L.I.D）原则是什么？逐一解释，并演示如何在实验练习和作业中使用这些原则。需要 C++ 代码示例。

35. 解释开放-封闭原则。用 C++ 示例演示 "开放-封闭原理"。

36. 解释利斯科夫置换原理（LSP）。使用 UML，举例说明：
    a. 违反 LSP
    b. 不违反《保密协议

    解释违反 LSP 的后果。用 C++ 代码演示。

37. 讨论各种散列方法，包括每种方法的优缺点。

38. 什么是哈希算法中的碰撞解决？描述两种碰撞解决算法。

39. 编写一个面向对象的 C++ 程序，以满足以下要求：

    数据文件 *data.txt* 包含按日期排序的气象数据行。数据文件可能包含重复条目。每一行都有一个日期（格式：日/月/年），然后是当天的平均环境空气温度（摄氏度）、当天的平均风速（千米/小时）和当天的平均太阳辐射（瓦/米^2）。天气数据之间用空白分隔。格式示例如下。虽然示例中只显示了 3 行数据（其中有 1 行重复），但可能有很多行数据代表了几十年的天气数据。重复数据也有可能没有按日期排序。它可能出现在数据文件的任何位置。

05/01/2008 19 10 560
05/01/2008 19 10 560
06/01/2008 25 7 800

编写一个主程序，将数据读入二叉搜索树 (BST)。您需要

您可以编写自己的 BST 以及程序所需的任何其他数据结构。BST 中不会存储重复数据。数据加载完毕后，用户可以使用键盘输入日期，然后程序会在屏幕（控制台）上打印天气数据。

由于数据量可能很大，因此搜索特定日期的效率必须很高。搜索指定日期数据的效率如何？请解释您的程序在效率方面的所有考虑（不仅仅是搜索）。您可能会有比较数值的代码。这些比较是否有效？

解释您的解决方案是如何设计以适应模型-视图-控制器 (**MVC**) 模式的。您的解释需要明确指出哪些类、子程序（函数、过程）适合 MVC 的哪个部分。您还应解释为什么它们适合 MVC 组件。

40. 什么是 PIMPL 成语？什么时候会用到这个成语？

41. 解释 "信息隐藏"。为什么要使用这个想法？使用 C++ 代码示例，演示如何实现信息隐藏。

42. 讨论 "信息隐藏 "与抽象之间的关系。使用 UML 和 C++ 示例来说明你的答案。

43. 什么是智能指针？

44. 使用伪代码编写一个*高效*算法，将两个**预先排序的**数组合并，并将其作为输入参数提供给算法。合并算法创建并返回一个新数组，其中包含两个原始数组中的所有值，所有值均已排序。该算法不修改原始数组，并且允许在生成的数组中出现重复值。如果不允许有重复值，该算法会有什么变化？

45. 在名为 *merge 的* C++ 模板子程序中实现合并算法。模板合并例程必须能够合并任何有序数据类型的排序数组。

46. 使用 C++ 编写一个名为 merge 的*高效*子程序。该子程序合并两个预先排序的向量，这两个向量作为输入参数提供给 *合并*子程序。*合并*子程序创建并返回一个新向量，其中包含两个原始向量中的所有值，所有值均已排序。该子程序不会修改原始向量，生成的向量中允许有重复值。

    解释为什么该算法是高效的。

提供整数测试数据，以测试合并子程序。在编写测试程序时，测试数据将存储在向量中，以测试您的合并例程。本题不需要编写测试程序，也不需要将测试数据放入向量中。您需要提供选择测试数据的理由。

47. 合并排序算法使用了什么算法策略，其效率如何（Big- O）？使用伪代码编写合并排序算法。

48. 解释集合数据结构。常见的集合操作有哪些？写出*有效的*

每种常见的集合操作的算法。要使算法有效，需要哪些前提条件？请解释你的答案。

49. 为每个设置操作编写测试计划。

50. 使用 C++ 实现每种集合算法，以执行集合运算。

51. 在 set1、set2 和 set3 均为 Set 类对象的情况下，编写有效的 *Set* union 算法的伪代码，以实现以下操作：

```
set3 = set1.union(set2);
```

要使算法有效，需要哪些前提条件？请解释你的答案。使用 C++，以名为 *union*

*的*模板函数实现 UNION 算法。

如果这三个集合的参数是
*工会，工会*现在是一个非朋友、非成员的功能?

52. 解释二进制搜索算法的使用。该算法的前提条件是什么，前提条件如何影响运行时性能（Big-O）。用 C++ 实现该算法。

53. 使用 C++，以子程序的形式实现二进制搜索算法。二进制搜索应该适用于任何**有序数据类型**的数组。使用由 Date 对象组成的有序向量测试搜索例程。矢量和日期与您在实验室和作业中创建的相同。

54. 使用图表和 C++ 代码解释以下内容：值参数、指针参数、引用参数。如果在参数上使用关键字 *const*，其设计意图是什么?

检查下面的方法原型。从设计意图或目的的角度来看，每个*制约*因素意味着什么?请酌情使用图表。

```
const int& methodA(const Vector& v, int * const * p) const;
```

55. 给定下面的子程序原型，用图表确定并解释内存的哪一部分是 "只读 "的。说

明在赋值语句的左侧不能出现哪个 `ptr` 或 `ptr` 去引用。

```
void f1 (int const * const * const ptr);
```

56. 使用 C++ 代码示例解释基类和抽象基类的区别。什么时候会用到这两个类？在 UML 图中显示这些类。

57. 用 C++ 示例解释什么是纯虚拟方法。在哪些软件设计情况下，你会使用具有纯虚方法的类？为你在这些情况下的使用辩护。

58. 使用 C++ 代码示例解释 "德墨忒尔法则"。

59. 使用 C++ 代码示例演示违反 "德墨忒尔法则 "的行为。对于本问题，您应使用您在实验和/或作业中完成的作业。如果您的实验/作业没有违反该法则，您可以使用其他示例，但您需要声明（就本问题而言）您从未违反过该法则。

    解释为什么这些都是违法行为，以及违法行为的后果。

60. 什么是 STL？

    STL 有一个堆栈数据结构。STL 中还有哪些其他数据结构？要使用某些 STL 数据结构，你需要编写哪些代码？要回答代码问题，请想一想要在主程序中使用 STL std::stack 需要编写哪些代码。

61. 使用 C++ 代码示例，解释迭代与递归的优缺点比较。

62. 在递归算法中，基数和通例之间有什么关系？

63. 用适当的例子解释算法分析中的下列术语：

    a. 算法的运行时间
    b. 最长运行时间
    c. 运行时间的增长速度
    d. 渐近上限

64. 按增长率升序排列下列 Big-O 值：O(n)、O(1)、$O(n^3)$、$O(n^2)$、$O(2^n)$、O(2n)、O(nlog n)、O(log n)。

65. 将未排序的数据插入向量、链表和二叉搜索树中，这样每个数据结构都有相同的数据。这些数据结构的搜索/查找算法的阶数（Big-O）分别是多少？请解释您的答案。

66. 为了尽量减少空间和时间复杂性，您必须存储大量数据，在决定使用何种数据结构时，您会进行哪些考虑？您在选择时会如何权衡？

67. 在复杂性分析中会遇到以下术语。请解释每个术语。

a. NP
b. P
c. 确定性算法
d. 非确定性算法

68. 什么是软件设计模式?

使用 C++ 代码示例解释*策略*设计模式。说明这种设计模式的用途，以及使用这种模式有什么好处。

使用 UML 设计一个使用*策略*设计模式的完整程序。

使用 C++ 实现你的设计。使用代码注释，突出显示使用该模式的地方。

69. 使用 UML 解释 "模型-视图-控制器"（**MVC**）模式。您的解释需要明确指出哪些类、子程序（函数、过程）适合 MVC 的哪个部分。您还应解释为什么它们适合 MVC 组件。

    以 C++ 程序的形式实现你的设计，并在代码注释中强调代码的哪些部分与 MVC 模式的哪些部分相匹配。

70. 有一条规则是 "除无行为聚合体外，数据成员应为私有"。回顾一下你的实验和作业，解释一下你在哪些地方违反了这条规则，在哪些地方遵守了这条规则。用你自己的代码来说明你的答案。在回答这个问题时，请强调你的类设计在哪些地方假装封装了行为。

71. 讨论是否有必要制定一条规则 "考虑让虚拟函数成为非公有函数，让公有函数成为非虚拟函数"。

72. 解释非虚拟接口（NVI）模式。

73. 解释什么是循环复杂性，以及它与程序逻辑复杂性的关系。循环复杂性与测试用例数量有何关系。提供 C++ 代码示例来说明你的答案--使用你自己在实验室和作业中的作品。