

## PAAT-C++中级 2 练习题四

### 一. 单项选择题 (每题 3.5 分, 共 20 题, 共 70 分)

1. C++中, 用于声明一个类的成员函数为虚函数的关键字是:
  - A. public
  - B. protected
  - C. virtual
  - D. Private
2. 下列哪个运算符用于检查指针是否为 NULL?
  - A. !
  - B. ~
  - C. nullptr
  - D. ==
3. C++中, 用于在类中声明静态成员变量的关键字是:
  - A. static
  - B. const
  - C. volatile
  - D. Inline
4. C++中, 使用 new 操作符分配的内存, 应当使用哪个操作符释放?
  - A. delete
  - B. free
  - C. release
  - D. Destroy
5. 下列哪个关键字用于声明一个类的成员函数为静态函数?
  - A. static
  - B. const
  - C. volatile
  - D. Virtual
6. C++中, 用于声明一个类的构造函数没有返回类型, 以下说法正确的是:
  - A. 构造函数可以返回任何类型
  - B. 构造函数可以返回 int 类型
  - C. 构造函数可以返回 void 类型
  - D. 构造函数没有返回类型
7. C++中, 用于声明一个模板类的关键字是:
  - A. template

- B. class
- C. struct
- D. Typedef

8. C++中，用于声明一个枚举类型的关键字是：

- A. enum
- B. union
- C. struct
- D. class

9. C++中，用于声明一个类的析构函数的关键字是：

- A. destructor
- B. ~
- C. delete
- D. destructor()

10. C++中，用于在函数中返回当前对象引用的关键字是：

- A. this
- B. self
- C. current
- D. Object

11. 下列哪个关键字用于声明一个类的成员变量为私有（private）访问权限？

- A. public
- B. private
- C. protected
- D. friend

12. C++中，用于声明模板的关键字是？

- A. template
- B. class
- C. struct
- D. typedef

13. 在 C++中，用于动态分配内存的运算符是？

- A. new
- B. malloc
- C. allocate
- D. reserve

14. C++中，用于释放由 new 分配的单个对象的内存的运算符是？

- A. delete
- B. free

- C. deallocate
- D. clear

15. 在 C++ 中，用于声明抽象类的关键字是？

- A. abstract
- B. virtual
- C. final
- D. abstract 不是 C++ 的关键字

16. 下列哪个运算符用于比较两个对象是否相等？

- A. ==
- B. =
- C. <=
- D. >=

17. 在 C++ 中，用于在类中声明一个成员函数为虚函数的关键字是什么？

- A. virtual
- B. final
- C. override
- D. Abstract

18. 下列哪个运算符用于在 C++ 中实例化一个对象？

- A. new
- B. delete
- C. &
- D. \*

19. 在 C++ 中，用于释放动态分配的内存的运算符是什么？

- A. new
- B. free
- C. delete
- D. Release

20. 在 C++ 中，用于声明一个成员变量为私有（private）的关键字是什么？

- A. public
- B. protected
- C. private
- D. Friend

二、编程题（每题 15 分，共 2 题，共 30 分）

21. 编写程序实现功能如下：

计算矩形面积。已知矩形的面积为长\*宽。在一行输入两个正整数（用空格分隔），分别代表矩形的长和宽，如果输入的数字包含零或负数，则直接输出“N”。

样例：

样例 1：

输入：3 4

输出：12

样例 2：

输入：-3 4

输出：N

22. 将长度为 n 的数组中第一个数移到数组末尾，其余数据依次往前平移一个位置

1 2 3 4 5 6 7 8 9 10

变为 2 3 4 5 6 7 8 9 10 1

输入 n ( $n \leq 100$ ) 和 n 个数，实现上述过程。