商业秘密※启用前

子悦汉化组开发者笔试试卷

考试时间：31 天　总分：150 分

考试须知：

1. **诚信考试，严禁抄袭作弊。**参考参考答案或使用工具辅助的，无权参与竞赛。
2. 参与竞赛请将答案写在 A4 白纸上，需回答所有题目，**包括选做题**。
3. 祝各位考生考试顺利！

一、单选题（每题 3 分，共 20 题，共 60 分）

1. nullptr 关键字的主要作用是？
   A. 表示空字符串
   B. 替代 NULL，明确表示空指针
   C. 用于初始化智能指针
   D. 表示未初始化的变量
2. 以下哪个是合法的基于范围的 for 循环？
   A. `for (int i = 0; i < vec.size(); ++i)`
   B. `for (auto& item : vec)`
   C. `for (item in vec)`
   D. `for (vec.begin() to vec.end())`
3. 关于 auto 关键字，错误的是？
   A. `auto x = 5;`
   B. `auto* p = new int(10);`
   C. `auto func(int a) { return a; }`
   D. `auto list = {1, 2, 3};`
4. std::unique_ptr 的特点是？
   A. 允许多个指针共享所有权
   B. 独占所有权，不可复制
   C. 需要手动调用 delete
   D. 自动进行引用计数
5. 以下 lambda 表达式语法正确的是？
   A. `[]() { return 0; }`
   B. `(x) -> { return x; }`
   C. `lambda x: x + 1`
   D. `[] => 42`
6. override 关键字的作用是？
   A. 允许重载函数

B. 显式标记重写虚函数

C. 禁止函数被继承

D. 优化函数调用

7. 以下初始化 std::vector 的方式正确的是？

A. std::vector<int> v{1, 2, 3};

B. std::vector v = [1,2,3];

C. std::vector<> v(3);

D. std::vector<int> v = new std::vector(3);

8. 关于异常处理，错误的是？

A. 使用 try/catch 块捕获异常

B. noexcept 表示函数不抛出异常

C. catch (Exception& e)可捕获所有异常

D. 派生类异常应优先被捕获

9. std::move 执行后，源对象的状态是？

A. 被立即销毁

B. 处于有效但未定义状态

C. 自动变为 nullptr

D. 与原对象完全独立

10. constexpr 变量必须在何时初始化？

A. 运行时动态计算

B. 编译期间确定值

C. 程序启动时初始化

D. 延迟到第一次使用时

11. 关于 C++20 引入的 char8_t 类型，以下描述正确的是：

A. 用于表示 UTF-16 编码的宽字符

B. 在 C++17 中已作为扩展类型存在

C. u8"字符串"的字面量类型在 C++20 中从 const char*变为 const char8_t*

D. 可直接隐式转换为 char 类型以兼容旧代码

12. 以下哪个智能指针支持引用计数？

A. std::unique_ptr

B. std::shared_ptr

C. std::weak_ptr

D. std::auto_ptr

13. 关于表达式 [=]() mutable { ... }，以下描述正确的是：

A. 捕获外部变量为只读

B. 按引用捕获外部变量

C. 允许修改按值捕获的变量

D. 必须包含返回类型声明

14. std::move 的作用是：

A. 转移对象的所有权

B. 将左值转换为右值引用

C. 执行深拷贝操作

D. 强制对象内存重定位

15. 以下关于 constexpr 的描述，错误的是：

A. 可用于编译时计算

B. 可以修饰函数返回值

C. 修饰的变量必须在编译期初始化

D. 不能用于修饰类成员函数

16. 以下代码的输出是：

```
int x = 0;

auto f = [x]() mutable { return ++x; };

std::cout << f() << f() << x;
```

A. 110

B. 111

C. 120

D. 121

17. 关于范围 for 循环，以下代码正确的是：

A. for (int i : 10) {...}

B. for (auto& item : std::vector{1,2,3}) {...}

C. for (const auto&& x : container) {...}

D. for (int i=0; auto x : vec) {...}

18. 以下哪个特性**不属于** C++17？

A. 结构化绑定（Structured Binding）

B. std::optional

C. 模块（Modules）

D. if constexpr

19. 关于右值引用（Rvalue Reference），声明正确的是：

A. int&& x = 5;

B. int& x = std::move(y);

C. const int&& x = y;

D. int&& x = y;

20. 以下代码可能引发的风险是：

```
std::shared_ptr<int> p1(new int(10));

std::shared_ptr<int> p2 = p1;

p1.reset(new int(20));
```

A. 内存泄漏
B. 空悬指针
C. 双重释放
D. 无风险

二、不定项选择题（每题 6 分，共 5 题，共 30 分）

选出你认为正确的答案，答案可能只有一个，也有可能有多个，**多选、不选、错选均不给分，少选给 3 分**。

1. 关于移动语义，正确的有：
   A. 移动构造函数应标记为 noexcept
   B. std::string 支持移动语义
   C. 移动后源对象必须置为 nullptr
   D. 可通过 std::move 触发移动操作
2. 关于 std::unique_ptr，正确的描述是：
   A. 支持自定义删除器
   B. 可以通过拷贝构造函数复制
   C. 可以转换为 std::shared_ptr
   D. 默认使用 delete 释放资源
3. 关于完美转发（Perfect Forwarding），正确的有：
   A. 依赖万能引用（Universal Reference）
   B. 需要配合 std::forward 使用
   C. 保证参数的值类别（Value Category）不变
   D. 只能用于模板函数
4. 以下属于类型安全特性的有：
   A. enum class
   B. static_cast
   C. dynamic_cast
   D. reinterpret_cast
5. 以下代码的正确输出是：

```
std::tuple<int, string> t(1, "test");

auto& [a, b] = t;

a = 2;

std::cout << get<0>(t);
```

A. 1
B. 2
C. 编译错误
D. 未定义行为

三、判断题（每题 1 分，共 10 题，共 10 分）

1. std::move 会转移对象的所有权。
2. constexpr 函数在 C++14 中允许函数体内包含循环和条件语句。
3. [=]() { x++; } 可以修改按值捕获的变量 x。
4. std::unique_ptr 可以通过拷贝构造函数共享所有权。
5. decltype(auto) 会保留表达式的值类别（值/引用）。
6. noexcept 函数中抛出异常会导致编译错误。
7. std::array<int, 5> 的大小可以在运行时动态调整。
8. std::shared_ptr 的引用计数是线程安全的。
9. if constexpr 在运行时根据条件选择代码分支。
10. std::optional<int> 可以表示一个可能不存在的 int 值。

四、主观题（共 3 题，共 50 分）

1. 默写《蜀道难》（10 分）
2. 解释下面的 C++ 代码（要求：解释运行逻辑，并给出运行结果）（20 分）

```
template<typename T>
void func(T&& t) {
    std::cout << t << "\n";
}
template<typename T, class... Args>
void func(T&& t, Args&&... args) {
    std::cout << t << "\n";
    func(args...);
}
int main() {
    func(123, 456.0, "what");
}
```

3. 已知函数 $f(x)=a^x+x^2-x\ln a$ $(a>0,\ a\neq 1)$.（30 分）
   1) 当 $a=e$ 时，求函数 $f(x)$ 的单调区间.（10 分）
   2) 若 $\exists x_1,x_2\in[-1,1]$，使得 $|f(x_1)-f(x_2)|\geq e-1$，求实数 a 的取值范围.（20 分）

五、选做题（共 1 题，共 50 分）

假如你是子悦汉化组的成员 ZiYue，你的朋友 EasyT_T 要用 C# 写代码，请你**用尖酸刻薄的方式**写一篇英文信劝导 EasyT_T 放弃 C#，使用 C++，内容包括：

1. C# 通常比 C++ 慢，因为它运行在 .NET 框架上，需要垃圾回收机制。
2. C# 的垃圾回收机制虽然方便，但也意味着开发者对内存管理的控制较少。
3. C# 代码需要依赖于特定的编译器和运行时环境，这会限制其灵活性。

要求：英文写作，不少于 100 词。

子悦汉化组开发者笔试试卷

参考答案

一、单选题

1-B 2-B 3-C 4-B 5-A 6-B 7-A 8-C 9-B 10-B 11-C 12-B 13-C 14-B 15-D 16-A 17-B 18-C 19-A 20-D

二、不定项选择题

1-ABD 2-ACD 3-ABC 4-AC 5-B

三、判断题

1-× 2-√ 3-× 4-× 5-√ 6-× 7-× 8-√ 9-× 10-√

四、主观题

第一题

白送 1 分，每段 3 分，错字扣 1 分。

第二题

1. 写对运行结果：5 分。
2. 正确解释函数实现：5 分。
   参数 1 分，模板 2 分，万能引用 2 分（右值引用则为 1 分）。
3. 正确解释解包：5 分。
   提到解包 1 分，解包结果什么样 2 分，参数 t 和 args 都是哪些 2 分。
4. 正确解释两个 func 函数的调用时间：5 分。

运行结果：三行输出，分别为：123、456.0 和 what。（5 分）

这段代码有两个函数模板（2 分），上面一个接受一个任意类型的万能引用（2 分）；下面一个则接受一个任意类型的万能引用，以及一个任意类型的可变参数万能引用。（2 分）（1 分）

main 函数是入口。调用 func 函数并传入三个参数。第一次调用时会调用下面一个 func，传入 123 作为参数 t，传入 456.0 和”what”作为可变参数 args。（2 分）进入 func 函数后，参数 t 被输出，args 被解包（1 分），456.0 作为 t，”what”作为可变参数 args 递归传给函数自己。（2 分）

第二次调用时，参数 t 为 456.0，被输出。此时可变参数解包，只有一个”what”，调用上面一个 func 函数（5 分），传入 t 并输出，递归结束。

第三题

（1）

代入 $a=e$ 得 $f(x)=e^x+x^2-x\ln e$，求导得 $f'(x)=2x+e^x-1$.（3 分）

令 $f'(x)=0$，得 $x=0$。画图可知：（5 分）

x	$(-\infty,0)$	0	$(0,+\infty)$
$f'(x)$	<0	0	>0
$f(x)$	单调递减	极小值	单调递增

因此：$(-\infty,0)$*上增*，$(0,+\infty)$*上减*.（2 分）

（2）

$\because\ \exists x_1,x_2\in[-1,1]$，*使得* $|f(x_1)-f(x_2)|\geq e-1$

$\therefore |f(x)_{max}-f(x)_{min}|\geq e-1$ （1 分）

对 $f(x)$ 求导得：$f'(x)=\ln a\,(a^x-1)+2x$.（1 分）

当 $a>1$ 时：（2 分）

$x<0$ 时 $a^x-1<0$，$\ln a>0$，$2x$*增* $\Longrightarrow f'(x)<0\Longrightarrow f(x)$*减*.

$x\geq 0$ 时 $a^x-1\geq 0,\ln a\geq 0,2x$*增* $\Longrightarrow f'(x)>0\Longrightarrow f(x)$*增*.

当 $0<a<1$ 时：（2 分）

$x<0$ 时 $a^x-1>0,\ln a<0,2x$*增* $\Longrightarrow f'(x)<0\Longrightarrow f(x)$*减*.

$x\geq 0$ 时 $a^x-1<0,\ln a<0,2x$*增* $\Longrightarrow f'(x)>0\Longrightarrow f(x)$*增*.

注意到：$[-1,0)\searrow(0,1]\nearrow$ （1 分）

带入 0 得到 $f(x)_{min}=f(0)=1$.

$$f(-1)=\frac{1}{a}+1+\ln a\,,f(1)=a+1-\ln a$$

如果：$f(1)-f(-1)>0\Longrightarrow f(1)>(f-1)\Longrightarrow f(x)_{max}=f(1)$ （1 分）

所以：$f(1)-f(-1)=a-\frac{1}{a}-2\ln a>0$ （1 分）

注意到：a >1 时不等式成立. （1 分）

$|f(x)_{max}-f(x)_{min}|\geq e-1\Leftrightarrow a-\ln a\geq e-1$ （1 分）

令 $g(x)=a-\ln a$，求导得 $g'(x)=1-\frac{1}{a}$，可得：$(0,1)\searrow(1,+\infty)\nearrow$ （2 分）

可得：$a\geq e$ 时不等式成立。（1 分）

当 $0<a<1$ 时 $f(x)_{max}=f(-1)$，可得 $|f(x)_{max}-f(x)_{min}|\geq e-1 \Leftrightarrow \frac{1}{a}-\ln\frac{1}{a}\geq e-1$.（2 分）

令 $\frac{1}{a}=t, t\in(1,+\infty)$，可得 $t\geq e \Longrightarrow \frac{1}{a}\geq e \Longrightarrow a\leq \frac{1}{e}$ （2 分）

综上：$a\in(-\infty,\frac{1}{e})\cup(e,+\infty)$.（2 分）

五、选做题

1. **不是英文不给分；不写标题扣 10 分。**
2. **少于 100 词最多 30 分。**
3. 文章必须包括的三条内容：每条 10 分。
4. 尖酸刻薄程度：10 分。
5. 语法：10 分。

An Intervention for Your Pathetic Coding Delusions

Dear EasyT_T,

Oh, look at you—clinging to C# like a toddler to a safety blanket. Let me enlighten your tragically misguided soul before you embarrass yourself further.

First, let's address the elephant in the room: C# is slower than a snail on sedatives. Why? Because it's busy babysitting your code on the .NET framework, complete with its adorable garbage collector. Sure, it's "convenient" to let the runtime clean up your messes, but don't you crave control? Or are you content letting Microsoft hold your hand while your code plods along like a sleepwalking sloth? C++ doesn't coddle you—it trusts you to manage memory like a grown-up. No training wheels, no excuses.

Ah, garbage collection—the crutch for developers who find memory management "too hard." How adorable. C#'s GC might spare you the horror of pointers, but it also robs you of precision. You're coding in a padded cell, EasyT_T. Meanwhile, C++ developers are out here sculpting memory like Michelangelos, crafting code that doesn't stutter every time the runtime decides to tidy up your incompetence.

And let's not forget C#'s hilarious dependency on .NET. You're shackled to Microsoft's ecosystem, praying their compiler doesn't sneeze on your code. C++? It runs anywhere—bare metal, embedded systems, even

your grandma's toaster. It's the Swiss Army knife of languages, while C# is a plastic spork at a five-star restaurant.

So, EasyT_T, are you a developer or a lab rat in Microsoft's cage? Swap out your training wheels for a sports car. Or keep wallowing in mediocrity—your funeral.

Yours in secondhand embarrassment,

ZiYue