yangzhy22022

#A. Checklist for C++ Beginners

客观题

> NOTE:
>
> 该比赛已结束，您无法在比赛模式下递交该题目。您可以点击"在题库中打开"以普通模式查看和递交本题。

Checklist for C++ Beginners

Answer the following questions according to the C++17 standard.

You may find the answers to these questions in any way, e.g. from the slides, from the Internet, from textbooks, from ChatGPT, or talk with your friends. **Some of you may even enumerate all possible answers and try them on OJ to find out the answers, which we highly advise against but cannot prevent.** But make sure you really have understood them. Similar questions may appear in exams, in code you write and in other materials you read. These are very basic things about C++, which average students should understand.

The OJ may not save your answers. Make sure you have saved them elsewhere, or you may have to re-answer all the questions for every submission.

1. (5 points) What is the type of the string literal `"this"`?

Hint: Type it in your code editor and put your mouse on it. Any modern code editor will tell you its type.

const char[5]

2. (5 points) Which standard library file is `std::cin` defined in?

- ○ A. `std::iostream`
- ◉ B. `iostream`
- ○ C. `iostream.h`
- ○ D. `std.iostream`

- ○ A. `string.h`
- ○ B. `cstring`
- ◉ C. `string`
- ○ D. `std::string`

4. (5 points) Let `p` be a pointer and `a` be of type `int [5]`. Which of the following expressions yield(s) an **lvalue**?

- ☑ A. `*p`
- ☐ B. `&a`
- ☑ C. `a[3]`
- ☑ D. `*&a`
- ☑ E. `++p`

5. (5 points) Which of the following statements regarding C++ IOStream is/are true?

- ☐ A. `std::cin` is a function used for reading things from input.
- ☐ B. When we use `std::cin` to read things, we need to pass the address of the variable.
- ☑ C. Unlike `printf`, `std::cout` is able to detect the type of the variable automatically and to choose the correct way to print its value.
- ☐ D. When using `std::cin` to read an integer variable, we need to make sure there are no leading whitespaces.

6. (5 points) Let `s` be an object of type `std::string`. What is the length of it?

- ☑ A. `s.size()`
- ☐ B. `std::strlen(s)`
- ☐ C. `sizeof(s)`
- ☐ D. `sizeof(s) / sizeof(char)`

7. (5 points) Read the following code.

Copy

```
#include <string>
int main() {
  std::string s1;
  std::string s2("Hello");
  std::string s3();
  std::string s4(48, 49);
  // ...
}
```

Which of the following statements is/are true?

- ☐ A. `s1` is uninitialized and has indeterminate value.
- ☑ B. `s1` represents the empty string `""`.
- ☐ C. The last character in `s2` is `'\0'`.

8. (5 points) Which of the following expressions will print `Hello world`? yangzhy22022

- [ ] A. `std::cout << "Hello " + "world"`
- [x] B. `std::cout << std::string("Hello") + " world"`
- [x] C. `std::cout << std::string("Hello") + ' ' + "world"`
- [ ] D. `std::cout << "Hello" + ' ' + std::string("world")`

9. (5 points) Read the following code.

Copy

```
#include <vector>
#include <iostream>
int main() {
  int n;
  std::cin >> n;
  std::vector v;
  for (int i = 0; i != n; ++i) {
    int value; std::cin >> value;
    v[i] = value;
  }
  std::cout << v << std::endl;
  return 0;
}
```

Which of the following statements is/are **false**?

- [x] A. The type of `v` is `std::vector`.
- [x] B. The type of `v` is deduced to be `std::vector<int>`.
- [x] C. `v[i] = value;` will add an element of value `value` to the end of `v`.
- [x] D. If the input is `3 10 20 30`, the output will be `[10, 20, 30]`.

10. (5 points) Read the following code.

Copy

```
std::string &fun(std::string &str) {
  const std::string &s1 = str;
  return str;
  return s1;
}
```

Which of the following statements is/are true?

- [ ] A. In the parameter declaration `std::string &str`, `&` is the address-of operator that takes the address of `str`.
- [ ] B. `const std::string &s1 = str;` makes `s1` a copy of the string that `str` refers to.
- [ ] C. `return str;` causes undefined behavior if the return-value is used by the caller, because `str` is a local variable that will be destroyed when the function returns.
- [ ] D. `return s1;` has the same effect as `return str;`
- [x] E. `return s1;` causes compile-error.

11. (5 points) Which of the following is/are correct?

☐ A.

```
std::string *p = new std::string("Hello");
std::free(p);
```

Copy

This code deallocates the block of memory dynamically allocated by `new`, so it has no memory leaks and has no undefined behaviors.

☐ B.

```
std::string *p = static_cast<std::string *>(std::malloc(sizeof(std::string)));
std::string *q = new std::string;
```

Copy

Both `*p` and `*q` are empty strings (equal to `std::string("")`).

☐ C.

```
int *p = new int[n];
for (int i = 0; i != n; ++i)
  delete p + i;
```

Copy

The loop here has the same effect as `delete[] p;`, which correctly deallocates all blocks of memory allocated by `new int[n]`.

☑ D.

```
int *a = new int[n]{1, 2, 3};
std::cout << a[3];
delete[] a;
```

Copy

Suppose `n >= 4`. This code will output `0` and has neither memory leaks nor undefined behaviors.

12. (5 points) Read the following code. Suppose `sizeof(int) == 4` and `sizeof(void *) == 8`.

```
void print_array_cpp(int (&a)[10]);
void print_array_c(int b[10]);
int a[10];
int b[20];
int ival;
```

Copy

Which of the following statements is/are true?

☐ A. The `a` on the first line and the `a` on the third line refer to the same variable.

☑ B. Both `print_array_cpp(a)` and `print_array_c(a)` compile.

☐ C. Both `print_array_cpp(b)` and `print_array_c(b)` compile.

☑ D. `print_array_c(&ival)` compiles, while `print_array_cpp(&ival)` does not.

☑ E. In function `print_array_cpp`, `sizeof(a)` is equal to `40`.

13. (5 * 2 points) Given the overloaded functions: yangzhy22022

```
void fun(int &);
void fun(const int &);
void fun(double);
```

Select the results of overload resolution for the following calls to `fun`.

(a)

```
int ival = 42;
fun(ival);
```

- (●) A. `fun(int &)`
- ( ) B. `fun(const int &)`
- ( ) C. `fun(double)`
- ( ) D. Ambiguous call
- ( ) E. No matching function for this call

(b)

```
const int cival = 42;
fun(cival);
```

- ( ) A. `fun(int &)`
- (●) B. `fun(const int &)`
- ( ) C. `fun(double)`
- ( ) D. Ambiguous call
- ( ) E. No matching function for this call

(c) `fun(3.14f)`

- ( ) A. `fun(int &)`
- ( ) B. `fun(const int &)`
- (●) C. `fun(double)`
- ( ) D. Ambiguous call
- ( ) E. No matching function for this call

(d) `fun(42)`

- ( ) A. `fun(int &)`
- (●) B. `fun(const int &)`
- ( ) C. `fun(double)`

(e) `fun('a')` (Hard. Try this out on your computer.)

yangzhy22022

- ○ A. `fun(int &)`
- ◉ B. `fun(const int &)`
- ○ C. `fun(double)`
- ○ D. Ambiguous call
- ○ E. No matching function for this call

14. (5 points) Select the group of functions that constitute function overloading.

- ☐ A. `int fun(int)` and `void fun(signed int)`
- ☐ B. `void fun(int *)` and `void fun(int [10])`
- ☑ C. `void fun()` and `int fun(double)`
- ☑ D. `int fun(std::vector<int>)` and `int fun(std::vector<double>)`

15. (5 points) Read the following code.

```
int a = 42;
int &b = a;
int *c = &b;
int &d = b;
b = d;
```

Copy

Which of the following statements is/are true?

- ☐ A. `b` is the address of `a`.
- ☐ B. `c` is a pointer that points to the object `b`.
- ☑ C. `d` is a reference that is bound to `a`.
- ☐ D. `b = d;` makes `b` bound to `d`.

16. (5 points) Which of the following statements is/are true?

- ☑ A.

`std::vector<std::vector<double>>` is a valid type.

- ☐ B.

```
std::vector<std::vector<double>> matrix(n, m);
```

Copy

`matrix` is initialized to be with `n` rows and `m` columns. That is, `matrix` is a vector that contains `n` elements, each of which is a vector containing `m` `double`s.

- ☑ C.

```
std::vector matrix(n, std::vector(m, 0.0));
```

Copy

The type of `matrix` is deduced to be `std::vector<std::vector<double>>`.

```
std::vector matrix(n, std::vector(m, 0.0));
```

Copy

yangzhy22022

`matrix` is initialized to be with `n` rows and `m` columns, with each element initialized to `0.0`.

17. (5 points) Read the following code.

```
int ival = 42;
const int cival = 42;
const int &cref = ival;
int &ref = const_cast<int &>(cref);
const int &cref2 = cival;
int &ref2 = const_cast<int &>(cref2);
```

Copy

Which of the following statements is/are true?

- [ ] A. This code will generate a compile-error.
- [x] B. `ref` is a reference that is bound to `ival`.
- [ ] C. The `const_cast` on the fourth line is unnecessary: `int &ref = cref;` works just fine, because `cref` is actually bound to a non-`const` variable.
- [x] D. After this code, `++ref2;` compiles but results in undefined behavior.

18. (5 points) Which of the following statements is/are true?

- [x] A. `static_cast<double>(sum) / n` has the same effect as `(double)sum / n`.
- [x] B. `const_cast` is typically used to cast away low-level `const`ness.
- [ ] C. Compared to the C++-style *named casts* (`what_cast<type>(expr)`), the C-style cast (`(type)expr`) is better because it can be used to perform any cast.
- [x] D. Compared to the C-style cast, the C++-style named casts are better because they are explicit and distinguish the possibly dangerous casts from others clearly.

19. (5 points) Which of the following is/are true?

- [ ] A.

Given the overloaded functions

```
void fun(int);
void fun(int *);
```

Copy

`fun(NULL)` will call `fun(int *)` and pass the null pointer value to it.

- [ ] B.

To obtain a null pointer, `NULL` is preferable to `nullptr` because `NULL` is defined as `(void *)0`.

- [ ] C.

Both `#define NULL 0` and `#define NULL (10 - 10)` conform to the C++ standard.

- [x] D.

递交

yangzhy22022

Homework 5

✓ 已认领

查看作业

成绩表

我的最近递交记录

帮助

状态
已结束

题目
5

开始时间
2023-3-31 1:30

截止时间
2023-4-15 0:00

可延期
24 小时

状态

评测队列

开发

开源

API

支持

帮助

QQ 群