

1. Answer: Same address

Explanation: `p1 == p1` checks if the pointer `p1` is equal to itself, which is always true.

2. Answer: `typedef struct { int x; } MyStruct;`

Explanation: This is the correct syntax for using `typedef` with a `struct` in C++.

3. Answer: The `std` namespace contains C++ standard library components.

Explanation: The `std` namespace includes standard library functions, objects, and types in C++.

4. Answer: 10, 20

Explanation: The `print` method of `Point` outputs the values of `x` and `y`.

5. Answer: To manage memory for function calls and local variables.

Explanation: A stack frame stores function arguments, local variables, and return addresses during function calls.

6. Answer: `protected`

Explanation: The `protected` specifier allows access only to derived classes, not to outside code.

7. Answer: Cannot have member functions

Explanation: Structures in C cannot contain member functions, while C++ structures can.

8. Answer: Undefined behavior

Explanation: Dereferencing an uninitialized pointer leads to undefined behavior.

9. Answer: To prevent multiple inclusions of a file

Explanation: Header guards ensure a header file is included only once during compilation.

10. Answer: Compile-time error

Explanation: Functions cannot be overloaded solely based on return type.

11. Answer: It is shared among all objects of the class.

Explanation: A static member variable is shared by all instances of the class.

12. Answer: The value pointed to by ptr cannot be modified.

Explanation: The `const int* ptr` indicates that the value being pointed to is constant.

13. Answer: Generating unique names for each overloaded function

Explanation: Name mangling is used to create unique names for functions with the same name but different parameter lists.

14. Answer: Undefined behavior

Explanation: Dereferencing a NULL pointer is undefined behavior and often causes a runtime error.

15. Answer: 10

Explanation: A `const int*` prevents modification through the pointer, even if the value is changed directly.

16. Answer: "abc.h" searches in the project directory first.

Explanation: #include "abc.h" looks for the file in the current directory before searching in system paths.

17. Answer: `int *ptr;`

Explanation: This is the correct syntax for declaring a pointer to an integer.

18. Answer: 0

Explanation: Printing a NULL pointer (`ptr = NULL`) outputs 0.

19. Answer: Compilation error at Line 2

Explanation: `ptr = 30` tries to modify a value through a const int, which is not allowed.

20. Answer: Structures in C++ can have access specifiers.

Explanation: Unlike in C, C++ structures can have private, public, and protected access specifiers.