Multiple Choice Quiz: Pointers and Structures in C++

1. What is a pointer in C++?

   - A) A variable that stores a value
   - B) A variable that stores a memory address
   - C) A data structure
   - D) A function

2. What operator is used to get the address of a variable?

   - A) *
   - B) &
   - C) ->
   - D) .

3. What does the * operator do when used with a pointer?

   - A) Gets the address
   - B) Declares a pointer
   - C) Dereferences to access the value
   - D) Deletes memory

4. How do you initialize a null pointer in C++?

   - A) ptr = 0;
   - B) ptr = nullptr;
   - C) ptr = NULL;
   - D) Both B and C

5. Where is stack memory allocated in C++?

   - A) For dynamic variables
   - B) For local variables
   - C) For global variables
   - D) For pointers only

6. Which keyword allocates memory on the heap?

   - A) malloc
   - B) new
   - C) alloc
   - D) create

7. What must be done to prevent memory leaks in C++?

   - A) Use nullptr
   - B) Use delete to free memory
   - C) Use stack memory only
   - D) Initialize pointers

8. How do you free memory for a dynamically allocated array?

- A) delete ptr;
- B) delete[] ptr;
- C) free ptr;
- D) clear ptr;

9. What is a C++ structure?

   - A) A loop construct
   - B) A user-defined data type grouping variables
   - C) A pointer type
   - D) A memory allocation function

10. How do you access a structure member using a pointer?

    - A) .
    - B) ->
    - C) &
    - D) *

11. What is the output of this code? `int x = 5; int* ptr = &x; *ptr = 10; std::cout << x;`

    - A) 5
    - B) 10
    - C) Address of x
    - D) Error

12. Which memory type is managed automatically in C++?

    - A) Heap
    - B) Stack
    - C) Global
    - D) Dynamic

13. What is the purpose of setting a pointer to `nullptr` after deletion?

    - A) To allocate new memory
    - B) To prevent dangling pointers
    - C) To increase performance
    - D) To access the value

14. Which operator is used to access structure members for a non-pointer object?

    - A) ->
    - B) .
    - C) *
    - D) &

15. What is a use of pointers in C++?

    - A) Declaring variables
    - B) Dynamic memory allocation
    - C) Creating loops
    - D) Defining functions

16. What happens if you access a pointer after deleting its memory?

- A) Program runs normally
- B) Undefined behavior
- C) Memory is reallocated
- D) Pointer becomes nullptr

17. In a structure `struct Point { int x; int y; };`, how is `x` accessed for `Point p;`?

- A) p->x
- B) p.x
- C) *p.x
- D) &p.x

18. How do you dynamically allocate a structure in C++?

- A) new struct
- B) new StructureName
- C) malloc(StructureName)
- D) create StructureName

19. What is the output of this code? `struct Data { int val; }; Data* d = new Data; d->val = 7; std::cout << d->val;`

- A) 7
- B) Address of d
- C) Error
- D) 0

20. Why is heap memory useful in C++?

- A) It is faster than stack memory
- B) It persists beyond function scope
- C) It is automatically deallocated
- D) It limits memory usage

21. What is a dangling pointer?

- A) A pointer to valid memory
- B) A pointer to deallocated memory
- C) A null pointer
- D) A pointer to a structure

22. Which of the following correctly declares a pointer to a structure?

- A) struct Point ptr;
- B) Point* ptr;
- C) *Point ptr;
- D) &Point ptr;