1. Pointers are used primarily for:
   1. Declaring variables
   2. **Allocating memory**
   3. Looping structures
   4. Input/output operations
2. Which keyword is used to deallocate memory allocated to array by the "new" operator?
   1. **delete[]**
   2. dealloc
   3. delete
   4. destroy
3. Which type of pointer stores the address of another pointer?
   1. **Double pointer**
   2. Reference pointer
   3. Null pointer
   4. Void pointer
4. What is the output of the following code snippet?

int\* ptr = new int;

cout << ptr;

* 1. The value stored at the address pointed by ptr
  2. The address of ptr
  3. **The address of the dynamically allocated memory**
  4. The address of the variable declared using new

1. What is the purpose of the "delete" operator in C++?
   1. **To deallocate memory allocated by "new"**
   2. To release resources of an object
   3. To remove a variable from memory
   4. To free up space on the hard disk
2. Which of the following is an advantage of using smart pointers over raw pointers?
   1. Smart pointers are faster than raw pointers.
   2. **Smart pointers automatically deallocate memory**.
   3. Smart pointers can be used for multi-dimensional arrays.
   4. Smart pointers require less memory.
3. Which of the following is not a type of smart pointer in C++?
   1. Unique pointer
   2. Shared pointer
   3. Weak pointer
   4. **Dynamic pointer**
4. Which type of pointer cannot be reassigned to point to another memory address?
   1. Null pointer
   2. Void pointer
   3. **Constant pointer**
   4. Smart pointer
5. What happens if you delete a null pointer in C++?
   1. It causes a compilation error.
   2. It frees up memory.
   3. **It has no effect.**
   4. It releases resources of an object.
6. Which keyword is used to check if a pointer points to a valid memory location?
   1. null
   2. void
   3. **NULL**
   4. valid
7. What is the difference between "malloc" and "new" in C++?
   1. "new" is used for allocating arrays, while "malloc" is used for allocating single objects.
   2. **"new" calls the constructor, while "malloc" does not.**
   3. "malloc" allocates memory from the heap, while "new" allocates memory from the stack.
   4. "new" returns a void pointer, while "malloc" returns a typed pointer.
8. What does the "sizeof" operator return when used with a pointer?
   1. **The size of the pointer itself**
   2. The size of the data type pointed to
   3. The size of the memory block allocated
   4. The address of the pointer
9. Which of the following is not a valid method to deallocate memory allocated by "new"?
   1. Using the "delete" operator
   2. **Using the "free" function**
   3. Calling the destructor explicitly
   4. Assigning the pointer to nullptr
10. Which type of memory management involves allocating memory for objects and automatically deallocating it when they are no longer needed?
    1. Static memory management
    2. Dynamic memory management
    3. **Automatic memory management**
    4. Manual memory management
11. Which operator is used to dynamically allocate an array of objects?
    1. **new**
    2. alloc
    3. malloc
    4. array
12. What is a memory leak?
    1. When a pointer is not initialized
    2. When a pointer is deallocated before it is used
    3. **When memory is allocated but not deallocated**
    4. When a pointer points to an invalid memory location
13. Which type of pointer points to a function?
    1. **Function pointer**
    2. Method pointer
    3. Lambda pointer
    4. Callable pointer
14. Which of the following is not a valid way to allocate memory dynamically in C++?
    1. Using the "new" operator
    2. Using the "malloc" function
    3. **Using the "allocate" keyword**
    4. Using the "calloc" function
15. What is the output of the following code snippet?

int\* ptr = NULL;

if (ptr)

cout << "Valid pointer";

else

cout << "Invalid pointer";

* 1. Valid pointer
  2. **Invalid pointer**
  3. Compilation error
  4. Runtime error

1. Which type of pointer can store the address of any data type?
   1. **Void pointer**
   2. Any pointer
   3. Universal pointer
   4. Dynamic pointer
2. What happens if you delete a pointer twice in C++?
   1. It causes a compilation error.
   2. It frees up memory.
   3. It has no effect.
   4. **It leads to undefined behavior.**
3. Which operator is used to access the member of an object through a pointer?
   1. .
   2. **->**
   3. :
   4. ::
4. Which type of pointer can hold the address of a constant object?
   1. Constant pointer
   2. Immutable pointer
   3. ReadOnly pointer
   4. **Const pointer**
5. What is the output of the following code snippet?

int\* ptr = new int(5);

cout << \*ptr;

* 1. The value stored at the address pointed by ptr
  2. The address of ptr
  3. **The value 5**
  4. The address of the dynamically allocated memory

1. The incorrect statement with respect to dangling pointers is \_\_\_\_\_\_\_\_\_\_\_
   1. Pointer pointing to non-existent memory location is called dangling pointer
   2. When a dynamically allocated pointer references the original memory after it has been freed, a dangling pointer arises
   3. **If memory leak occurs, it is mandatory that a dangling pointer arises**
   4. Dangling pointer may result in segmentation faults and potential security risks
2. Garbage collector frees the programmer from worrying about \_\_\_\_\_\_\_\_\_\_\_
   1. Dangling pointers
   2. Creating new objects
   3. **Memory leak**
   4. Segmentation errors
3. Choose the right option string\* x, y;
   1. **x is a pointer to a string, y is a string**
   2. y is a pointer to a string, x is a string
   3. Both x and y are pointers to string types
   4. none of the above
4. Which from the following is not a correct way to pass a pointer to a function?
   1. Non-constant pointer to non-constant data
   2. A non-constant pointer to constant data
   3. A constant pointer to non-constant data
   4. **All of the above**
5. What does the following statement mean?

int (\*fp)(char\*)

* 1. Pointer to a pointer
  2. Pointer to an array of chars
  3. **Pointer to function taking a char\* argument and returns an int**
  4. Function taking a char\* argument and returning a pointer to int

1. Which of the following is illegal?
   1. int \*ip;
   2. string s, \*sp = 0;
   3. **int i; double\* dp = &i**
   4. int \*pi = 0;
2. What will happen in this code?

int a = 100, b = 200;

int \*p = &a, \*q = &b

p = q;

1. b is assigned to a
2. **p now points to b**
3. a is assigned to b
4. q now points to a