MCQ's in C++ Language

1. Which of the following is the correct syntax of including a user-defined header file in C++?

a) `#include "filename"`
b) `#include <filename>`
c) `#include [filename]`
d) `#include {filename}`

Answer: a
Explanation: The correct syntax to include a user-defined header file is `#include "filename"`, while system headers use angular brackets.

---

2. Which of the following is used to terminate a function in C++?

a) `return`
b) `exit()`
c) `abort()`
d) All of the above

Answer: d
Explanation: `return` is used to return control from a function, while `exit()` and `abort()` terminate the program execution.

---

3. What is the correct syntax for declaring an integer array of 10 elements?

a) `int arr[10];`
b) `int arr(10);`
c) `int[10] arr;`
d) `arr[10] int;`

Answer: a
Explanation: Arrays in C++ are declared using the syntax `type array_name[size];`.

---

4. Which of the following is a loop construct in C++?

a) `for`
b) `while`
c) `do-while`
d) All of the above

Answer: d
Explanation: C++ supports `for`, `while`, and `do-while` loops for iterative execution.

---

5. Which of the following can be used to define a constant in C++?

a) `#define`
b) `const`
c) Both `a` and `b`
d) None of the above

Answer: c
Explanation: Both `#define` (preprocessor directive) and `const` can be used to declare constants in C++.

---

6. What does the keyword `this` represent in C++?

a) A pointer to the current object
b) The first parameter of a function
c) A function return type
d) The last parameter of a function

Answer: a
Explanation: The `this` pointer holds the address of the current object and is used in object-oriented programming.

---

7. Which of the following operators cannot be overloaded in C++?

a) `+`
b) `->`
c) `::`
d) `[]`

Answer: c
Explanation: The scope resolution operator `::` cannot be overloaded in C++.

---

8. What is the size of a pointer in a 64-bit C++ environment?

a) 2 bytes
b) 4 bytes
c) 8 bytes
d) Depends on the data type

Answer: c
Explanation: In a 64-bit system, pointers typically have a size of 8 bytes regardless of the data type they point to.

---

9. How many types of polymorphism are supported in C++?

a) 1
b) 2
c) 3
d) 4

Answer: b
Explanation: C++ supports two types of polymorphism: compile-time (overloading and templates) and run-time (virtual functions).

---

10. What is the output of the following code?

```
cpp
Copy code
int x = 5;
cout << ++x;
```

a) 5
b) 6
c) 7
d) 0

Answer: b
Explanation: `++x` increments `x` by 1 before printing, so the output will be 6.

---

11. What is the purpose of a virtual function in C++?

a) To achieve dynamic binding
b) To make a function static
c) To define multiple main functions
d) To optimize memory usage

Answer: a
Explanation: Virtual functions are used to achieve dynamic binding, allowing derived class functions to override base class functions.

---

12. Which feature of C++ is used to implement function overloading?

a) Inheritance
b) Polymorphism
c) Encapsulation
d) Abstraction

Answer: b
Explanation: Function overloading is a type of compile-time polymorphism where functions with the same name can perform different tasks.

---

13. What is the default access specifier for members of a class in C++?

a) `public`
b) `private`
c) `protected`
d) `static`

Answer: b
Explanation: The default access specifier for class members is `private`.

---

14. How do you create a reference variable in C++?

a) `int &x = y;`
b) `int *x = &y;`
c) `int x = &y;`
d) `int &x = *y;`

Answer: a
Explanation: Reference variables are created using the `&` operator, where `x` refers to `y`.

---

15. What is the output of the following code?

```
cpp
Copy code
int x = 10;
cout << (x == 10 ? "True" : "False");
```

a) True
b) False
c) 10
d) Error

Answer: a
Explanation: The conditional operator checks if `x == 10`, and since the condition is true, the output is "True".

---

16. Which operator is used to access members of a structure in C++?

a) `.`
b) `->`
c) `*`
d) `::`

Answer: a
Explanation: The dot `.` operator is used to access members of a structure when using the object directly.

---

17. What is a destructor in C++?

a) A special function invoked when an object is created
b) A special function invoked when an object is destroyed
c) A special function invoked when a program starts
d) None of the above

Answer: b
Explanation: A destructor is a special member function that is called when an object goes out of scope or is explicitly deleted.

---

18. Which keyword is used to prevent a member function from being overridden?

a) `override`
b) `static`
c) `final`
d) `const`

Answer: c
Explanation: The `final` keyword prevents further overriding of a virtual function in derived classes.

---

19. What is a pure virtual function?

a) A function that does not have any definition
b) A function defined only in derived classes
c) A function with a body
d) A function that can be called without an object

Answer: a
Explanation: A pure virtual function is declared with `= 0` and must be overridden by derived classes.

---

20. What is the purpose of the `static` keyword in C++?

a) To allocate memory at runtime
b) To restrict variable visibility to the file
c) To maintain the value of a variable between function calls
d) Both `b` and `c`

Answer: d
Explanation: The `static` keyword can be used to restrict visibility of a variable to the file and maintain variable value between function calls.

---

21. How do you dynamically allocate memory for an array in C++?

a) `int* arr = new int[10];`
b) `int arr[10];`
c) `int arr = new int(10);`
d) `int arr = malloc(10);`

Answer: a
Explanation: The `new` keyword dynamically allocates memory for an array of size 10.

---

22. Which of the following is true about constructors in C++?

a) They have a return type
b) They are automatically called when an object is created
c) They can be virtual
d) None of the above

Answer: b
Explanation: Constructors are special functions that are automatically called when an object is instantiated.

---

23. What is the return type of the main function in C++?

a) `void`
b) `int`

c) `float`
d) `char`

Answer: b
Explanation: The `main` function returns an integer value indicating the success or failure of the program.

---

24. Which of the following statements about templates is correct?

a) Templates allow writing generic programs
b) Templates do not support classes
c) Templates can be instantiated only once
d) Templates must return `void`

Answer: a
Explanation: Templates enable the creation of generic classes and functions, allowing code to work with any data type.

---

25. What is a friend function in C++?

a) A function that can access private and protected members of a class
b) A function that belongs to more than one class
c) A function defined inside a class
d) A function that is private

Answer: a
Explanation: A friend function can access the private and protected members of the class in which it is declared as a friend.

---

26. What is the output of the following code?

```
int x = 3, y = 4;

cout << (x & y);
```

a) 7
b) 3
c) 0
d) 4

Answer: c
Explanation: The bitwise AND operation on `3` and `4` results in `0` because no bits are set in both numbers.

---

27. Which of the following containers is part of the Standard Template Library (STL)?

a) `vector`
b) `list`
c) `map`
d) All of the above

Answer: d
Explanation: The Standard Template Library (STL) includes containers like `vector`, `list`, and `map`.

---

28. Which of the following is not a type of inheritance in C++?

a) Single inheritance
b) Multiple inheritance
c) Partial inheritance
d) Multilevel inheritance

Answer: c
Explanation: There is no concept of "partial inheritance" in C++. The valid types of inheritance are single, multiple, multilevel, hierarchical, and hybrid.

---

29. Which of the following correctly describes the use of `namespace` in C++?

a) To resolve name conflicts
b) To allocate memory
c) To implement inheritance
d) To perform type casting

Answer: a
Explanation: Namespaces are used to avoid name conflicts in large projects by grouping entities like classes, functions, and variables.

---

30. What does the term RAII (Resource Acquisition Is Initialization) mean in C++?

a) Memory is allocated before object creation
b) Resource is automatically managed through object lifecycle
c) An object must be explicitly deleted after use
d) None of the above

Answer: b
Explanation: RAII refers to the concept where resource management is tied to the lifetime of objects, ensuring that resources are automatically cleaned up.

31. Which of the following statements is correct about the `const` keyword in C++?

a) It prevents changes to the variable's value after initialization.
b) It can be applied only to primitive data types.
c) It is used to declare pointers only.
d) It restricts the usage of a variable globally.

Answer: a
Explanation: The `const` keyword is used to make a variable read-only after its initialization, meaning its value cannot be modified later.

---

32. What is the output of the following code?

```
int x = 10;
int* ptr = &x;
cout << *ptr;
```

a) The memory address of `x`
b) The value of `x`
c) 0
d) A compile-time error

Answer: b
Explanation: The dereference operator (`*ptr`) accesses the value of `x`, which is `10`.

---

33. What is the correct way to declare a function pointer in C++?

a) `int *f();`
b) `int (*f)();`
c) `int f*();`
d) `int &(f)();`

Answer: b
Explanation: The correct syntax for a function pointer is `int (*f)();` which means `f` is a pointer to a function that returns `int`.

---

34. Which of the following types of casting is the most dangerous and should be avoided when possible?

a) `static_cast`
b) `dynamic_cast`
c) `const_cast`
d) `reinterpret_cast`

Answer: d
Explanation: `reinterpret_cast` performs low-level casting and can lead to unexpected behavior or errors if used improperly. It simply reinterprets the bits of an object without regard for its type.

---

35. Which of the following correctly represents the syntax of a constructor in C++?

a) `void ClassName();`
b) `int ClassName();`
c) `ClassName();`
d) `ClassName(int x = 0);`

Answer: c
Explanation: A constructor in C++ does not have a return type, and its name must match the class name.

---

36. In C++, a class can inherit from multiple base classes. What is this feature called?

a) Multiple inheritance
b) Single inheritance
c) Hybrid inheritance
d) Hierarchical inheritance

Answer: a
Explanation: Multiple inheritance allows a class to inherit from more than one base class in C++.

---

37. What is the function of the `typeid` operator in C++?

a) It returns the type of a variable at runtime.
b) It allocates memory for a variable.
c) It deletes a pointer.
d) It compares two objects.

Answer: a
Explanation: `typeid` is used to determine the type of a variable or object at runtime, providing information about the type.

---

38. What is the output of the following code?

```
int arr[] = {1, 2, 3, 4};
cout << sizeof(arr)/sizeof(arr[0]);
```

a) 4
b) 16
c) 1
d) Undefined

Answer: a
Explanation: `sizeof(arr)` gives the total size of the array in bytes, and `sizeof(arr[0])` gives the size of one element. Dividing them gives the number of elements in the array, which is 4.

---

39. Which of the following is true about friend classes in C++?

a) Friend classes can access all private and protected members of another class.
b) Friend classes are subclasses of the class.
c) Friend classes inherit the base class members.
d) Friend classes cannot access private members.

Answer: a
Explanation: Friend classes can access all private and protected members of the class in which they are declared as friends.

---

40. How do you explicitly call a base class constructor in a derived class in C++?

a) `base_class();`
b) `base_class::base_class();`
c) `derived_class:base_class();`
d) `base_class(parameter_list);`

Answer: d
Explanation: To explicitly call a base class constructor in a derived class, you call it in the initializer list, like `derived_class::derived_class() : base_class(parameter_list) {}`.

---

41. What does the `delete` operator do in C++?

a) Frees memory allocated by `new`.
b) Deletes an object file.
c) Compiles the program.
d) Deletes all files in a directory.

Answer: a
Explanation: The `delete` operator deallocates memory that was previously allocated with the `new` operator.

---

42. What will be the output of the following code?

```
cpp
Copy code
int a = 10, b = 20;
a = a ^ b;
b = a ^ b;
a = a ^ b;
cout << a << " " << b;
```

a) 10 20
b) 30 10
c) 20 10
d) 0 0

Answer: c
Explanation: This is a common bitwise XOR swap trick. After the three XOR operations, the values of `a` and `b` are swapped.

---

43. Which of the following is the correct syntax for defining a destructor in C++?

a) `~ClassName();`
b) `!ClassName();`
c) `ClassName!();`
d) `ClassName::~();`

Answer: a
Explanation: Destructors in C++ have the same name as the class, preceded by a tilde `~`. They are called when an object is destroyed.

---

44. Which of the following statements is false about inline functions in C++?

a) They are expanded at the point of invocation.
b) They are useful for short, frequently used functions.
c) They always reduce the code size.
d) The compiler can ignore the `inline` request.

Answer: c
Explanation: Inline functions may increase code size because the function code is duplicated at each point where it is invoked.

---

45. In C++, what is the default constructor?

a) A constructor that initializes all data members to 0.
b) A constructor that has no parameters.
c) A constructor that is called only once.
d) A constructor that returns the class object.

Answer: b
Explanation: A default constructor is a constructor with no parameters. If not defined explicitly, the compiler generates one automatically.

---

46. What is the role of the `new` operator in C++?

a) It allocates memory dynamically on the stack.
b) It allocates memory dynamically on the heap.
c) It creates local variables.
d) It allocates memory statically.

Answer: b
Explanation: The `new` operator allocates memory dynamically on the heap, which must later be deallocated using `delete`.

---

47. Which of the following is not a valid C++ data type?

a) `int`
b) `char`
c) `float`
d) `real`

Answer: d
Explanation: `real` is not a valid C++ data type. The correct data type for floating-point numbers is `float` or `double`.

---

48. How can you prevent a function from being overridden in C++?

a) Use the `private` keyword.
b) Use the `static` keyword.
c) Use the `final` keyword.
d) Use the `const` keyword.

Answer: c
Explanation: The `final` keyword prevents a function from being overridden by derived classes.

---

49. Which of the following correctly describes exception handling in C++?

a) It handles compile-time errors.
b) It handles runtime errors.
c) It guarantees program termination.
d) It avoids using the stack.

Answer: b
Explanation: Exception handling in C++ is used to handle runtime errors and allows the program to recover from unexpected events.

---

50. Which of the following is not a keyword in C++?

a) `throw`
b) `explicit`
c) `typename`
d) `include`

Answer: d
Explanation: `include` is a preprocessor directive, not a keyword. The `#include` directive is used to include files.

---

51. What does the `continue` statement do in C++?

a) It terminates the loop.
b) It skips the current iteration and moves to the next iteration.
c) It repeats the current iteration.
d) It exits the program.

Answer: b
Explanation: The `continue` statement skips the remaining code in the current loop iteration and moves to the next iteration.

---

52. Which of the following best describes the use of a virtual destructor?

a) It allows polymorphic deletion of objects.
b) It prevents object creation.
c) It can only be used with static functions.
d) It automatically calls the base class destructor.

Answer: a
Explanation: A virtual destructor ensures that the correct destructor is called for an object, especially when using polymorphism (i.e., base class pointers to derived objects).

---

53. What is the output of the following code?

```
int x = 10;
cout << (x % 3 == 0 ? "Divisible" : "Not Divisible");
```

a) Divisible
b) Not Divisible
c) 0
d) Error

Answer: b
Explanation: Since `10 % 3` is not equal to `0`, the output is "Not Divisible".

---

54. Which of the following operators is used to resolve the scope of a global variable in C++?

a) `::`
b) `.`
c) `->`
d) `&`

Answer: a
Explanation: The scope resolution operator `::` is used to specify that a variable belongs to the global scope, even if a local variable with the same name exists.

---

55. In C++, how do you explicitly free memory allocated by the `new` operator?

a) `delete variable;`
b) `free(variable);`
c) `remove(variable);`
d) `clear(variable);`

Answer: a
Explanation: The `delete` operator is used to deallocate memory that was allocated using `new`.

---

56. What is the output of the following code?

```
int a = 5, b = 6;
cout << (a > b ? a : b);
```

a) 5
b) 6
c) 0
d) Error

Answer: b
Explanation: Since `b` is greater than `a`, the output will be `6`.

---

57. Which of the following is true about the `std::string` class in C++?

a) It is a primitive data type.
b) It can be used like a normal array.
c) It supports dynamic memory allocation.
d) It does not support concatenation.

Answer: c
Explanation: `std::string` dynamically manages memory and supports operations like concatenation and substring extraction.

---

58. What does the `throw` keyword do in C++?

a) It defines an exception handler.
b) It terminates the program.
c) It raises an exception.
d) It rethrows a caught exception.

Answer: c
Explanation: The `throw` keyword is used to raise an exception, which can be caught by a `try-catch` block.

---

59. What is the output of the following code?

```
int x = 5;
int& ref = x;
ref = 10;
cout << x;
```

a) 5
b) 10
c) Error
d) Undefined behavior

Answer: b
Explanation: Since `ref` is a reference to `x`, assigning `10` to `ref` changes the value of `x` to `10`.

---

60. Which of the following is not a feature of Object-Oriented Programming (OOP) in C++?

a) Encapsulation
b) Inheritance
c) Procedural abstraction
d) Polymorphism

Answer: c
Explanation: Procedural abstraction is a feature of procedural programming. Encapsulation, inheritance, and polymorphism are key features of OOP.