MCQs

1. Which access specifier specifies that the derived class has no access to the member?

A) public

B) private

C) protected

D) default

Answer: B) private

2. In C++, which type of inheritance is supported by default?

A) Single Inheritance

B) Multiple Inheritance

C) Multilevel Inheritance

D) Hierarchical Inheritance

Answer: A) Single Inheritance

3. In C++, can a derived class access private members of the base class?

A) Yes, always

B) Yes, but only if the derived class is a friend of the base class

C) No, never

D) Yes, but only if the base class is a friend of the derived class

Answer: C) No, never

4. In C++, which type of polymorphism is achieved by function overloading?

A) Compile-time polymorphism

B) Run-time polymorphism

C) Dynamic polymorphism

D) Static polymorphism

Answer: A) Compile-time polymorphism

5. Which keyword is used in a derived class to indicate that a member function is intended to override a virtual function in the base class?

A) override

B) virtual

C) extend

D) over

Answer: A) override

6. Which type of polymorphism is achieved by function overriding?

A) Compile-time polymorphism

B) Run-time polymorphism

C) Dynamic polymorphism

D) Static polymorphism

Answer: B) Run-time polymorphism

7. In C++, what is the purpose of a pure virtual function?

A) To make the function inaccessible

B) To define a default implementation for the function

C) To declare a function without providing implementation

D) To make the function static

Answer: C) To declare a function without providing implementation

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

A) +

B) ::

C) .

D) =

Answer: B) ::

9. What is the return type of the overloaded assignment operator = in C++?

A) void

B) int

C) The class type being assigned

D) bool

Answer: C) The class type being assigned

10. What is the purpose of the overloaded stream insertion operator << in C++?

A) To read input from the user

B) To display output to the console

C) To concatenate strings

D) To perform addition of objects

Answer: B) To display output to the console

11. Which of the following cannot be a constructor?

A. Parameterized constructor

B. Copy constructor

C. Destructor

D. Static constructor

Answer: C. Destructor

12. Which of the following statements about constructors is true?

A. Constructors have a return type.

B. Constructors can have arguments.

C. Constructors cannot be overloaded.

D. Constructors can be called explicitly.

Answer: B. Constructors can have arguments.

13. Which of the following statements is true about destructors?

A. Destructors have a return type.

B. Destructors can be overloaded.

C. Destructors cannot be inherited.

D. Destructors can be called explicitly.

Answer: B. Destructors can be overloaded.

14. How many destructors can a class have?

A. Only one

B. Two

C. It depends on the number of objects created.

D. None

Answer: A. Only one

15. Can a constructor be virtual?

A. Yes

B. No

C. It depends on the compiler.

D. Only in C++11 and later.

Answer: B. No

16. What is the syntax for creating an object with a parameterized constructor?

A. ClassName objectName();

B. ClassName objectName = ClassName();

C. ClassName objectName = ClassName(value);

D. ClassName objectName(value);

Answer: D. ClassName objectName(value);

17. Which constructor is called when an object is passed as an argument to a function by value?

A. Default constructor

B. Parameterized constructor

C. Copy constructor

D. Destructor

Answer: C. Copy constructor

18. Which type of constructor cannot have a return type?

A. Default constructor

B. Parameterized constructor

C. Copy constructor

D. Destructor

Answer: D. Destructor

19. Which type of type conversion does a dynamic\_cast perform?

A. Implicit conversion

B. Explicit conversion

C. Runtime conversion

D. Compile-time conversion

Answer: C. Runtime conversion

20. Which of the following is an example of implicit type conversion?

A. static\_cast

B. dynamic\_cast

C. int x = 5.6;

D. int x = int(5.6);

Answer: C. int x = 5.6;