C++ TEST-14 (POLYMORPHISM)

Total points 50/50 ?

STUDENT NAME *

Mayank

✓ 1. What does polymorphism mean in C++? * 1/1

○ A) Many functions

● B) Many forms ✓

○ C) Many variables

○ D) Many classes

✓ 2. How many types of polymorphism are there in C++? * 1/1

○ A) 1

● B) 2 ✓

○ C) 3

○ D) 4

✓ 3. **Which are the two types of polymorphism in C++?** * 1/1

○ A) Compile-time and Run-time

○ B) Static and Dynamic

● C) Both A and B ✓

○ D) None

✓ 4. **Function overloading is an example of:** * 1/1

● A) Compile-time polymorphism ✓

○ B) Run-time polymorphism

○ C) Dynamic binding

○ D) Operator overriding

✓ 5. **Function overriding is an example of:** * 1/1

○ A) Compile-time polymorphism

● B) Run-time polymorphism ✓

○ C) Static binding

○ D) Operator overloading

✓ 6. **Which keyword is used to achieve runtime polymorphism?** * 1/1

○ A) static

● B) virtual ✓

○ C) protected

○ D) dynamic

✓ 7. **In function overloading, functions:** * 1/1

○ A) Have same name and parameters

● B) Have same name but different parameters ✓

○ C) Have different names

○ D) Must be virtual

✓ 8. **Operator overloading is an example of:** * 1/1

● A) Compile-time polymorphism ✓

○ B) Run-time polymorphism

○ C) Dynamic binding

○ D) None

✓ 9. **Function overriding occurs between:** * 1/1

○ A) Functions in same class

● B) Functions in base and derived classes ✓

○ C) Constructors

○ D) Destructors

✓ 10. **Which keyword ensures a function cannot be overridden?** * 1/1

● A) final ✓

○ B) static

○ C) protected

○ D) sealed

✓ 11. **Which function call is resolved at compile time?** * 1/1

○ A) Virtual function

● B) Overloaded function ✓

○ C) Overridden function

○ D) Pure virtual function

✓ 12. **Which function call is resolved at run time?** * 1/1

● A) Virtual function ✓

○ B) Static function

○ C) Inline function

○ D) Friend function

✓ 13. **What is the mechanism used to resolve a function call at runtime?** * 1/1

○ A) Static binding

● B) Dynamic binding ✓

○ C) Early binding

○ D) Compile binding

✓ 14. **Which of the following supports compile-time polymorphism?** * 1/1

○ A) Function overloading

○ B) Operator overloading

● C) Both A and B ✓

○ D) Virtual function

✓ 15. **Which of the following supports runtime polymorphism?** * 1/1

● A) Virtual functions ✓

○ B) Inline functions

○ C) Friend functions

○ D) Constructors

✓ 16. **To achieve runtime polymorphism, what is required?** * 1/1

● A) Virtual function and inheritance ✓

○ B) Function overloading

○ C) Operator overloading

○ D) Friend function

✓ 17. **What is a virtual function?** * 1/1

○ A) A static function

● B) A function that can be overridden in derived class ✓

○ C) A friend function

○ D) A private function

✓ 18. **What is the purpose of a virtual function?** * 1/1

○ A) To overload functions

● B) To achieve runtime polymorphism ✓

○ C) To define templates

○ D) To protect data

✓ 19. **Which keyword is used to declare a virtual function?** * 1/1

● A) virtual ✓

○ B) override

○ C) polymorph

○ D) dynamic

✓ 20. **What is a pure virtual function?** * 1/1

● A) Function with no body ✓

○ B) Function with definition

○ C) Static function

○ D) Friend function

✓ 21. **How do you declare a pure virtual function?** * 1/1

● A) virtual void show() = 0; ✓

○ B) void show();

○ C) virtual show();

○ D) void show() = virtual;

✓ 22. **A class containing a pure virtual function is called:** * 1/1

● A) Abstract class ✓

○ B) Virtual class

○ C) Base class

○ D) Derived class

✓ 23. **Can an abstract class have constructors?** * 1/1

● A) Yes ✓

○ B) No

✓ 24. **Can an abstract class be instantiated?** * 1/1

○ A) Yes

● B) No ✓

✓ 25. **Which function cannot be virtual?** * 1/1

● A) Constructor ✓

○ B) Destructor

○ C) Member function

○ D) Overridden function

✓ 26. **What happens if a derived class does not override a pure virtual function?** * 1/1

○ A) No error

● B) Derived class also becomes abstract ✓

○ C) Compilation success

○ D) Function ignored

✓ 27. **Can destructors be virtual?** * 1/1

● A) Yes ✓

○ B) No

✓ 28. **Why should destructors be virtual?** * 1/1

● A) To free resources properly when deleting derived objects ✓

○ B) To increase performance

○ C) To make class abstract

○ D) None

✓ 29. **What is the output when a base class pointer points to a derived class object?** *1/1

○ A) Base class version

● B) Derived class version (if function is virtual) ✓

○ C) Both

○ D) Error

✓ 30. **Which binding does virtual function use?** * 1/1

○ A) Early binding

● B) Late binding ✓

○ C) Static binding

○ D) None

✓ 31. **Which type of polymorphism uses function signatures?** * 1/1

● A) Compile-time polymorphism ✓

○ B) Runtime polymorphism

○ C) Dynamic binding

○ D) None

✓ 32. **What does operator overloading allow?** * 1/1

● A) Redefining operators for user-defined types ✓

○ B) Using multiple operators together

○ C) Creating new operators

○ D) Overriding existing classes

✓ 33. **What is the main advantage of polymorphism?** * 1/1

○ A) Code reusability

○ B) Code readability

● C) Flexibility and maintainability ✓

○ D) Execution speed

✓ 34. **Which function is automatically virtual in base class?** * 1/1

● A) Destructor ✓

○ B) Constructor

○ C) Static

○ D) Inline

✓ 35. **Which of the following can't be overloaded?** * 1/1

● A) :: ✓

○ B) +

○ C) ==

○ D) []

✓ 36. **Overloading functions based on return type alone is:** * 1/1

○ A) Allowed

● B) Not allowed ✓

✓ 37. **Which operator cannot be used to overload member functions?** * 1/1

○ A) new

○ B) =

● C) . (dot) ✓

○ D) ->

✓ 38. **Which keyword allows a derived class to explicitly override a base virtual function?** * 1/1

○ A) virtual

● B) override ✓

○ C) final

○ D) super

✓ 39. **Can static functions be virtual?** * 1/1

○ A) Yes

● B) No ✓

✓ 40. **Which concept allows a single interface to represent different data types?** * 1/1

○ A) Abstraction

● B) Polymorphism ✓

○ C) Encapsulation

○ D) Overloading

✓ 41. **Virtual functions must be:** * 1/1

○ A) Defined outside class

● B) Member functions of class ✓

○ C) Static functions

○ D) Global functions

✓ 42. **Which of the following is not necessary for runtime polymorphism?** * 1/1

○ A) Inheritance

○ B) Virtual function

● C) Static function ✓

○ D) Base class pointer

✓ 43. **Which keyword ensures no derived class can override a function?** * 1/1

● A) final ✓

○ B) stop

○ C) static

○ D) override

✓ 44. **What does dynamic_cast do in polymorphism?** * 1/1

● A) Converts base pointer to derived pointer safely ✓

○ B) Converts string to integer

○ C) Performs static conversion

○ D) Converts primitive types

✓ 45. **What is vtable in C++?** * 1/1

● A) Table of virtual function addresses ✓

○ B) Data structure for objects

○ C) Operator overload table

○ D) None

✓ 46. **What is vptr in C++?** * 1/1

○ A) Virtual function pointer

● B) Pointer to vtable ✓

○ C) Object pointer

○ D) None

✓ 47. **Which polymorphism is achieved without pointers or references?** * 1/1

● A) Compile-time polymorphism ✓

○ B) Runtime polymorphism

○ C) Both

○ D) None

✓ 48. **Virtual functions in C++ must have:** * 1/1

● A) Same name and parameters in base and derived ✓

○ B) Different parameters

○ C) Same name only

○ D) None

✓ 49. **Can constructors participate in polymorphism?** * 1/1

● A) No ✓

○ B) Yes, if virtual

✓ 50. **What is the key benefit of runtime polymorphism?** * 1/1

○ A) Faster execution

● B) Code flexibility during execution ✓

○ C) Compile-time safety

○ D) None