

## Đề ôn tập

### Lập trình hướng đối tượng (OOP)

- Tính chất nào có ở Class nhưng không có ở Structure?
  - Có Data member
  - Có Member function**
  - Cho phép có Static data
  - Có từ khóa truy cập Public
- Tính năng nào của lập trình hướng đối tượng minh họa khả năng sử dụng lại mã nguồn?
  - Polymorphism
  - Abstraction
  - Encapsulation
  - Inheritance**
- Tính chất nào của lập trình hướng đối tượng là sai trong C++?
  - Có thể viết Code mà không sử dụng Class
  - Code phải chứa ít nhất 1 class**
  - Một class phải có các hàm thành viên
  - Nên có ít nhất 1 object được khai báo trong chương trình
- Tập tiêu đề (header file) nào cần có trong C++ để có thể lập trình hướng đối ?
  - iostream.h
  - stdio.h
  - stdlib.h
  - OOP có thể được sử dụng mà không cần header file nào.**
- Tính đa hình (polymorphism) trong OOP nghĩa là gì?
  - Khái niệm cho phép ghi đè hàm**
  - Khái niệm để ẩn dữ liệu
  - Là khái niệm cho phép giữ mọi thứ trong các modul khác nhau.
  - Là khái niệm cho phép giữ mọi thứ trong cùng một đơn vị.
- C++ là \_\_\_\_\_
  - ngôn ngữ lập trình thủ tục
  - ngôn ngữ lập trình hướng đối tượng
  - ngôn ngữ lập trình hàm
  - là ngôn ngữ hỗ trợ cả lập trình thủ tục và lập trình hướng đối tượng**
- Tính năng nào của OOP không được sử dụng trong đoạn code C++ sau?

```
class A
{
    int i;
public:
    void print() {cout<<"hello"<<i;}
}

class B: public A
{
```

```
int j;  
public:  
void assign(int a){j = a;}  
}
```

- a) Abstraction
  - b) Encapsulation
  - c) Inheritance
  - d) Polymorphism
8. Phương pháp tiếp cận nào được sử dụng trong C++?
- a) Left-right
  - b) Right-left
  - c) Bottom-up
  - d) Top-down
9. Đây là tên biến đặt đúng trong C++?
- a) VAR\_1234
  - b) \$var\_name
  - c) 7VARNAME
  - d) 7var\_name
10. Toán tử nào dùng để kiểm tra kích thước của một đối tượng?
- a) sizeof(objectName)
  - b) size(objectName)
  - c) sizeofobject(objectName)
  - d) sizedobject(objectName)
11. Chỉ định phạm vi truy cập mặc định của các dữ liệu và hàm trong một Lớp khi không được khai báo tường minh trong C++ là :?
- a) Private
  - b) Protected
  - c) Public
  - d) Tùy thuộc trình biên dịch.
12. Kích thước của 1 class là \_\_\_\_\_
- a) Tổng kích thước của các biến khai báo trong Class
  - b) Tổng kích thước của tất cả các biến cùng với các biến được kế thừa trong Class
  - c) Kích thước của biến lớn nhất trong Class
  - d) Class không có kích thước
13. Class nào có thể có hàm thành viên mà không kèm phần cài đặt của?
- a) Class mặc định
  - b) String class
  - c) Template class
  - d) Lớp ảo (Abstract class).
14. Điều nào sau đây là sai đối với một hàm thành viên của một class?
- a) Tất cả hàm thành viên phải được định nghĩa.
  - b) Hàm thành viên có thể được định nghĩa bên trong hay bên ngoài phần thân của class
  - c) Hàm thành viên không cần phải khai báo bên trong phần định nghĩa của lớp
  - d) Hàm thành viên có thể là hàm bạn của một lớp khác bằng cách sử dụng từ khóa friend.

15. Phát biểu nào chính xác với đoạn mã nguồn sau?

```
class student
{
    int marks;
    public : student ()
    {
        cout<<"New student details can be added now";
    }
};
student s1;
```

- a) Cout không thể được sử dụng bên trong hàm tạo
- b) Hàm tạo chỉ được chứa các khởi tạo
- c) Đoạn code trên chạy bình thường
- d) Đoạn code trên phát sinh lỗi

16. Cú pháp nào cho định nghĩa lớp là sai?

- a) class student{ };
- b) student class{ };
- c) class student{ public: student(int a){ } };
- d) class student{ student(int a){} };

17. Định nghĩa nào mô tả tốt nhất một đối tượng?

- a) là một thể hiện của một lớp
- b) là một thể hiện của chính nó
- c) là con của một lớp
- d) là tổng quan của một lớp

18. Từ khóa nào được sử dụng để xác định các hàm thành viên tĩnh?

- a) static
- b) stop
- c) open
- d) state

19. Từ khóa nào được sử dụng để khai báo hàm ảo?

- a) virtual
- b) virt
- c) anonymous
- d) virtually

20. Có bao nhiêu đối tượng có thể được khai báo với mỗi lớp trong một chương trình?

- a) 32768
- b) 127
- c) 1
- d) Không có quy định nào

21. Các hàm ảo không thể được tạo với tính chất của

- a) Hàm tĩnh
- b) Hàm có tham số
- c) Hàm có tham số mặc định
- d) Hàm không có tham số

22. Đoạn code nào sau đây đúng?

- a) `class student{ }s1,s2; s1.student()=s2.student();`
- b) `class student{ }s1; class topper{ }t1; s1=t1;`
- c) `class student{ }s1,s2; s1=s2;`
- d) `class student{ }s1; class topper{ }t1; s1.student()=s2.topper();`

23. Phát biểu nào đúng với đoạn code sau?

```
class A
{
    private : int marks; char name[20];
    public :
    A(int x=100)
    {
        marks=x;
    }
};
```

- a) Các đối tượng có thể được tạo với 1 hoặc 0 tham số
- b) Đối tượng chỉ có thể được tạo với một tham số
- c) Đối tượng có thể được tạo với nhiều tham số
- d) Đối tượng chỉ có thể được tạo mà không có tham số

24. Câu lệnh nào là đúng để khởi tạo các đối tượng của lớp student cho dưới đây?

```
class student{
int marks;
int cgpa;
public: student(int i, int j){
    marks=i;
    cgpa=j
}
};
```

- a) `student s[3]={ s(394, 9); s(394, 9); s(394,9); };`
- b) `student s[2]={ s(394,9), s(222,5) };`
- c) `student s[2]={ s1(392,9), s2(222,5) };`
- d) `student s[2]={ s[392,9], s2[222,5] };`

25. Điều kiện nào sau đây không cần thiết cho hàm tạo của lớp?

- a) Hàm tạo phải cùng tên với lớp.
- b) Hàm tạo không có kiểu giá trị trả về
- c) Hàm tạo phải có định nghĩa phần thân hàm
- d) Hàm tạo có thể có tham số

26. Đối tượng nào sẽ được tạo ra trước?

```
class student
{
    int marks;
};
student s1, s2, s3;
```

- a) s1 rồi đến s2 rồi đến s3
- b) s3 rồi đến s2 rồi đến s1
- b) s2 rồi đến s3 rồi đến s1
- c) Tất cả được tạo ra cùng một thời điểm.

27. Khi nào một hàm hủy được gọi?

- a) Sau khi kết thúc vòng đời của đối tượng.
- b) Bất cứ lúc nào trong quá trình sử dụng đối tượng.
- c) Vào cuối toàn bộ chương trình
- d) Ngay trước khi kết thúc vòng đời của đối tượng

28. Làm sao để phân biệt hàm tạo và hàm hủy?

- a) Hàm hủy có kiểu trả về nhưng hàm tạo thì không
- b) Lập trình viên không thể xác định các hàm hủy, nhưng có thể xác định các hàm tạo
- c) Các hàm hủy có ký hiệu dấu ngã (~) đặt phía trước tên hàm còn hàm tạo thì không
- d) Hàm hủy có cú pháp giống với hàm tạo

28. Hàm khởi tạo nào sẽ được gọi từ đối tượng obj2 trong chương trình C ++ sau đây?

```
class A
{
    int i;
    A()
    {
        i=0;
    }
    A(int x)
    {
        i=x+1;
    }
    A(int y, int x)
    {
        i=x+y;
    }
};
A obj1(10);
A obj2(10,20);
A obj3;
```

- a) A(int y, int x)
- b) A(int y; int x)
- c) A(int y)
- d) A(int x)

29. Đáp án nào dưới đây là đại diện cho 1 hàm tạo đúng cú pháp?

- a) -classname()
- b) classname()

- c) ()classname
- d) ~classname()

30. Điều gì xảy ra nếu các thành viên không tĩnh được sử dụng trong hàm thành viên tĩnh?

a) Thực thi bình thường

b) Lỗi thời gian biên dịch

c) Thực thi nếu chức năng thành viên đó không được sử dụng

d) Lỗi khi chạy chương trình

31. Điều nào sau đây xác định thừa kế đơn?

a) Một lớp cơ sở dẫn xuất một lớp khác

b) Một lớp dẫn xuất kế thừa từ một lớp cơ sở

c) Một lớp cơ sở kế thừa từ một lớp dẫn xuất

d) Một lớp dẫn xuất dẫn xuất từ một lớp dẫn xuất khác

31. Nếu sử dụng kế thừa đơn, chương trình sẽ bao gồm \_\_\_\_\_

a) Có ít nhất 2 lớp

b) Nhiều nhất 2 lớp

c) Đúng 2 lớp

d) Nhiều nhất 4 lớp

32. Ngôn ngữ nào không hỗ trợ kế thừa đơn?

a) Java

b) C ++

c) Kotlin

d) Tất cả các ngôn ngữ đều hỗ trợ

33. Phương thức khởi tạo nào sẽ được gọi đầu tiên từ các lớp liên quan đến kế thừa đơn từ đối tượng của lớp dẫn xuất?

a) Hàm tạo lớp cơ sở

b) Phương thức khởi tạo lớp dẫn xuất

c) Cả hai hàm tạo lớp tại một thời điểm

d) Lỗi thời gian chạy

34. Các thành phần thành viên không tham gia kế thừa sẽ được khai báo là \_\_\_\_\_

- a) Thành viên public
- b) Thành viên protected
- c) Thành viên private
- d) Các thành viên private hoặc protected

35. Trong đoạn code sau lớp nào có thể minh họa được tính đa hình:

```
abstract class student
{
    public : int marks;
    calc_grade();
}
class topper:public student
{
    public : calc_grade()
    {
        return 10;
    }
};
class average:public student
{
    public : calc_grade()
    {
        return 20;
    }
};
class failed{ int marks; };
```

- a) Chỉ class student thể hiện tính đa hình
  - b) Các class student, topper cùng nhau thể hiện tính đa hình
  - c) Tất cả class student, topper và average cùng thể hiện tính đa hình
  - d) Class failed cũng nên kế thừa class student để code có thể thể hiện tính đa hình
36. Chương trình dưới đây khi chạy sẽ hiển thị gì ra màn hình?

```
class student
{
    public : int marks;
    void disp()
    {
        cout<<"its base class"
    };
}
class topper:public student
{
    public :
    void disp()
    {
        cout<<"Its derived class";
    }
}
void main()
{
    student s; topper t;
    s.disp();
    t.disp();
}
```

```
}
```

- a) Its base classIts derived class
- b) Its base class Its derived class
- c) Its derived classIts base class
- d) Its derived class Its base class

37. Chương trình dưới đây khi chạy sẽ hiển thị gì ra ngoài màn hình?

```
#include <iostream>
using namespace std;
class A{
public:
    A() {
        cout<<"Constructor called\n";
    }
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main(int argc, char const *argv[])
{
    A *a = new A[5];
    delete[] a;
    return 0;
}
```

- a) Chỉ hiển thị "Constructor called" 5 lần
- b) Hiển thị "Constructor called" 5 lần và sau đó hiển thị "Destructor called" 5 lần
- c) "Constructor called" 5 lần và sau đó "Destructor called" duy nhất 1 lần
- d) Lỗi

38. Chương trình dưới đây khi chạy sẽ hiển thị gì ra ngoài màn hình?

```
1.  #include <iostream>
2.  using namespace std;
3.  class rect
4.  {
5.      int x, y;
6.      public:
7.      void val (int, int);
8.      int area ()
9.      {
10.         return (x * y);
11.     }
12. };
13. void rect::val (int a, int b)
14. {
15.     x = a;
16.     y = b;
17. }
18. int main ()
19. {
20.     rect rect;
21.     rect.val (3, 4);
```



```

22.         cout << "rect area: " << rect.area();
23.         return 0;
24.     }

```

a) rect area: 24

**b) rect area: 12**

c) Lỗi biên dịch vì bên trong có biến được sử dụng làm tên lớp và tên biến trong dòng # 20

d) rect area: 56

39. Chương trình dưới đây khi chạy sẽ hiển thị gì ra màn hình?

```

1.     #include <iostream>
2.     using namespace std;
3.     class MyInterface
4.     {
5.     public:
6.         virtual void Display() = 0;
7.     };
8.     class Class1 : public MyInterface
9.     {
10.    public:
11.        void Display()
12.        {
13.            int a = 5;
14.            cout << a;
15.        }
16.    };
17.    class Class2 : public MyInterface
18.    {
19.    public:
20.        void Display()
21.        {
22.            cout << " 5" << endl;
23.        }
24.    };
25.    int main()
26.    {
27.        Class1 obj1;
28.        obj1.Display();
29.        Class2 obj2;
30.        obj2.Display();
31.        return 0;
32.    }

```

a) 5

**b) 10**

**c) 5 5**

d) 15

40. Chương trình dưới đây khi chạy sẽ hiển thị gì ra ngoài màn hình?

```

#include <iostream>
#include <string>

```

```

using namespace std;
class A
{
    int a, b;
    float d;
public:
    void change(int i) {
        a = i;
    }
    void value_of_a() {
        cout<<a;
    }
};

class B: private A
{
};

int main(int argc, char const *argv[])
{
    B b;
    cout<<sizeof(B);
    return 0;
}

```

- a) 8
- b) 12**
- c) Lỗi
- d) Lỗi phân đoạn. ( chết chương trình)

41. Chương trình dưới đây khi chạy sẽ hiển thị gì ra ngoài màn hình?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    float d;
public:
    int a;

```

```

        void change(int i){
            a = i;
        }
        void value_of_a(){
            cout<<a;
        }
};

class B: public A
{
    int a = 15;
public:
    void print(){
        cout<<a;
    }
};

int main(int argc, char const *argv[])
{
    B b;
    b.change(10);
    b.print();
    b.value_of_a();

    return 0;
}

```

- a) 1010
- b) 1510
- c) 1515
- d) 5110

42. Chương trình dưới đây khi chạy sẽ hiển thị gì ra ngoài màn hình?

```

1.  #include <iostream>
2.  using namespace std;
3.  int operate (int a, int b)
4.  {
5.      return (a * b);
6.  }
7.  float operate (float a, float b)
8.  {
9.      return (a / b);
10. }
11. int main ()
12. {
13.     int x = 5, y = 2;
14.     float n = 5.0, m = 2.0;
15.     cout << operate (x, y);
16.     cout << operate (n, m);
17.     return 0;
18. }

```

- a) 119
- b) 102.5

- c) 123.4
- d) 128.4

43. Chương trình dưới đây khi chạy sẽ hiển thị gì ra ngoài màn hình?

```
1.  #include <iostream>
2.  using namespace std;
3.  class Cat
4.  {
5.      public:
6.          int age;
7.          int weight;
8.  };
9.  int main()
10. {
11.     Cat f;
12.     f.age = 56;
13.     cout << "Gates is " ;
14.     cout << f.age << " years old.\n";
15. }
```

a) Gates is

b) Gates is 56 years old

c) Lỗi

d) Gates is 53 years old

44. Chương trình dưới đây khi chạy sẽ hiển thị gì ra màn hình?

```
1.  #include <iostream>
2.  using namespace std;
3.  class Rect
4.  {
5.      int x, y;
6.      public:
7.          void set_values (int,int);
8.          int area ()
9.          {
10.             return (x * y);
11.          }
12.     };
13.     void Rect::set_values (int a, int b)
14.     {
15.         x = a;
16.         y = b;
17.     }
18.     int main ()
19.     {
20.         Rect recta, rectb;
21.         recta.set_values (5, 6);
22.         rectb.set_values (7, 6);
23.         cout << "recta area: " << recta.area();
24.         cout << "rectb area: " << rectb.area();
25.         return 0;
```

26. }

a) recta area: 30 rectb area: 42

b) recta area: 20 rectb area: 34

c) recta area: 30 rectb area: 21

d) recta area: 30 rectb area: 33

45. Điều nào sau đây là đúng với các class được định nghĩa dưới đây:

```
class student
{
    int marks;
public: student() {}
    student(int x)
    {
        marks=x;
    }
};
main()
{
    student s1(100);
    student s2();
    student s3=100;
    return 0;
}
```

a) Đối tượng s3 được tạo

b) Chỉ S1 và S2 được tạo

c) Chương trình chạy và tất cả các đối tượng được khởi tạo

d) Chương trình lỗi khi biên dịch

46. Chương trình dưới đây khi chạy sẽ hiển thị gì ra ngoài màn hình?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    void change(int i){
        a = i;
    }
    void value_of_a(){
        cout<<a;
    }
};

int A::a = 5;

int main(int argc, char const *argv[])
{
    A a1 = A();
    A a2 = A();
}
```

```

A a3 = A();
a1.change(10);
a1.value_of_a();
a2.value_of_a();
a3.value_of_a();
return 0;
}

```

- a) 1055
- b) 555
- c) 101010**
- d) 51010

47. Chương trình dưới đây khi chạy sẽ hiển thị gì ra màn hình?

```

1.  #include <iostream>
2.  using namespace std;
3.  class number
4.  {
5.      int i;
6.      public:
7.      int geti();
8.      void puti(int j);
9.  };
10.      int number::geti()
11.      {
12.          return i;
13.      }
14.      void number::puti(int j)
15.      {
16.          i = j;
17.      }
18.      int main()
19.      {
20.          number s;
21.          s.puti(10);
22.          cout << s.geti( );
23.          return 0;
24.      }

```

- a) 10**
- b) 11**
- c) 20
- d) 22

48. Chương trình dưới đây khi chạy sẽ hiển thị gì ra màn hình?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    float d;
    public:

```

```

        virtual void func(){
            cout<<"Hello this is class A\n";
        }
};

class B: public A
{
    int a = 15;
public:
    void func(){
        cout<<"Hello this is class B\n";
    }
};

int main(int argc, char const *argv[])
{
    A *a = new A();
    B b;
    a = &b;
    a->func();
    return 0;
}

```

- a) Hello this is class A
- b) Hello this is class B**
- c) Hello this is class A Hello this is class B
- d) Lỗi phân đoạn

50. Phát biểu nào dưới đây là không chính xác về hàm ảo?

- a) Chúng được dùng để thể hiện tính đa hình thời gian chạy
- b) Chúng được sử dụng để ẩn đi các đối tượng**
- c) Hàm ảo được khai báo bằng từ khoá virtual
- d) Tất cả những điều nêu trên.