

Câu hỏi ôn tập C++

Câu 1. Mảng tĩnh là gì?

- a) Là một chuỗi các phần tử cùng kiểu dữ liệu nằm liên tiếp trong bộ nhớ
- b) Là một chuỗi các phần tử
- c) Là một chuỗi các phần tử cùng kiểu dữ liệu không nằm liên tiếp trong bộ nhớ
- d) Không có câu trả lời nào đúng trong các câu trên

Câu 2. Khi khai báo lớp trong C++, hành vi của đối tượng được khai báo như là:

- A. Hàm.
- B. Tên lớp.
- C. Biến
- D. Không khai báo trong lớp.

Câu 3. Trong C++, khi khởi tạo một đối tượng thì hàm nào sau đây được thực hiện

- A. Hàm constructor (hàm tạo)
- B. Hàm destructor (hàm hủy)
- C. Hàm friend
- D. Hàm member

Câu 4. Kết quả chương trình sau:

```
class Test {  
  
};  
  
void main() {  
    Test t;  
}
```

- A. Lỗi thực thi
- B. Lỗi biên dịch.
- C. Chương trình thực thi mà không xuất gì hết.
- D. Chương trình chạy vô tận

Câu 5. Hãy xem xét đoạn mã sau:

```

class A{
    int a,b;

    public:
        float F1,F2;
};
class B:public A{
    ...
}

```

Hỏi: B sử dụng được các biến thành viên nào của A

- A. F1, F2
- B. a, b
- C. a,b,F1,F2
- D. Không sử dụng được biến thành viên nào

Câu 6. Một biến được gọi là biến toàn cục nếu:

- A. Nó được khai báo bên ngoài tất cả các hàm, ngoại trừ hàm main().
- B. Nó được khai báo bên ngoài tất cả các hàm kể cả hàm main().
- C. Nó được khai báo bên ngoài hàm main().
- D. Nó được khai báo bên trong hàm main().

Câu 7. Sử dụng cách truyền nào trong hàm sẽ không làm thay đổi giá trị của biến trong chương trình chính:

- A. Truyền bằng tham trị
- B. Truyền bằng giá trị địa chỉ của tham số
- C. Cả A. và B. đều đúng
- D. Cả A. và B. đều sai

Câu 8. Luật nào sẽ bị mất tác dụng với hàm bạn?

- a) Các thành viên private và protected của một lớp không thể được truy cập từ bên ngoài
- b) Thành viên private và protected có thể được truy cập từ bất kỳ nơi nào
- c) cả a và b
- d) Không có câu trả lời nào ở trên là đúng

Câu 9. Những phát biểu nào sau đây là đúng khi nói về lớp:

- A. Một đối tượng là một thể hiện (instance) cụ thể của một lớp.
- B. Một lớp là một thiết kế hay mẫu cho các đối tượng cùng kiểu
- C. Lớp định nghĩa các thuộc tính và các phương thức chung cho tất cả các đối tượng cùng một loại nào đó.
- D. Tất cả đều đúng

Câu 10. Khai báo class nào sau đây là hợp lệ

- A. public class A { }
- B. class B { }
- C. class A { int x; };
- D. object A { int x; };

Câu 11. Kết quả chương trình sau:

```
#include <iostream>

using namespace std;

class Test {
    public: int n;
    private: void Display(){cout<<n;}
    public: Test():n(15){}
};

void main() {
    Test t;
    t.Display();
}
```

- A. Lỗi thực thi
- B. Lỗi biên dịch.
- C. Chương trình thực thi xuất ra màn hình: 15.
- D. Chương trình thực thi mà không xuất gì ra màn hình.

Câu 12. Kết quả chương trình sau:

```
#include <iostream>
```

```

using namespace std;

class ABC {
    int n;

    public: ABC(int x):n(x){}

    void Print(){cout<<n;}
};

void main() {
    ABC t;

    t.Print();
}

```

- A. Lỗi thực thi
- B. Lỗi biên dịch.
- C. Chương trình thực thi xuất ra màn hình một số âm.
- D. Chương trình thực thi mà không xuất gì ra màn hình.

Câu 13. Kết quả chương trình sau:

```

#include <iostream>

using namespace std;

class Point {
    int xVal, yVal;

public:
    Point(int x = 0, int y = 0){
        xVal = x ; yVal = y ;
    }

    void Print(){
        cout<< " ("<<xVal <<","<<yVal<<" )";
    }
};

void main() {
    Point pt(5);
}

```

```
    pt.Print();  
}
```

A. Lỗi do khởi tạo đối tượng pt không đúng.

B. Hiển thị trên màn hình (5,5)

C. Hiển thị trên màn hình (0,5)

D. Hiển thị trên màn hình (5,0)

Câu 14. Kết quả chương trình sau

```
#include <iostream>  
  
using namespace std;  
  
class Point {  
    int xVal, yVal;  
  
public:  
    Point(int x = 0, int y = 0){  
        xVal = x ; yVal = y ;  
        cout<< "So nguyen " ;  
    }  
  
    Point(double x = 0, double y = 0){  
        xVal = x ; yVal = y ;  
        cout<< "So thuc " ;  
    }  
  
    void Print() ;  
};  
  
void Point :: Print(){ cout<< " ("<<xVal <<","<<yVal<<")"; }  
  
void main() {  
    Point pt(5.0);  
    pt.Print();  
}
```

A. Hiển thị trên màn hình So thuc (5,0)

B. Hiển thị trên màn hình So nguyen (5,5)

C. Hiển thị trên màn hình Số nguyên (5,0)

D. Chương trình bị lỗi biên dịch.

Câu 15. Kết quả chương trình sau:

```
#include <iostream>

using namespace std;

class Test {
    int t;
    public :
        Test(int t){ Test::t = t ; }
        void Print(){cout<<t;}
};

void main() {
    Test test(4);
    test.Test::Print();
}
```

A. Lỗi thực thi

B. Lỗi biên dịch.

C. Chương trình thực thi xuất ra màn hình : 4

D. Chương trình thực thi mà không xuất gì ra màn hình.

Câu 16. Kết quả chương trình là :

```
#include <iostream>

using namespace std;

class Teacher {
    static int n;
    public : Teacher (){ cout<<" "<<n++ ; }
};

int Teacher::n = 0;

void main() {
    Teacher t1;
```

```

    Teacher t2;

    Teacher t3;

    Teacher t4;
}

```

A. Xuất ra màn hình : 0 1 2 3

B. Xuất ra màn hình : 1 2 3 4

C. Xuất ra màn hình : 0 1 2 0

D. Xuất ra màn hình : 1 2 3 1

Câu 17. Kết quả chương trình là :

```

#include <iostream>

using namespace std;

class Point{
    private: int xVal, yVal;
    public: void Print(){
        cout<<"("<<xVal<<","<<yVal<<")";
    }
    Point(int x=0, int y=0):xVal(x),yVal(y){}
    Point(int x){
        Point::xVal = Point::yVal = x;
    }

    friend Point operator + (Point, Point);
};

Point operator + (Point p1, Point p2)
{
    return Point(p1.xVal+p2.xVal, p1.yVal+p2.yVal);
}

void main(){
    Point p1(3,4);
    Point p = p1+3;
}

```

```
    p.Print();  
}
```

- A. Lỗi biên dịch
- B. Lỗi thực thi
- C. Kết quả là (6,7)
- D. Kết quả là (6,4)

Câu 18. Kết quả chương trình là :

```
#include<iostream>  
  
using namespace std;  
  
class Base{  
    public:  
        Base(){ cout<<"Base class"<<endl; }  
};  
  
class Derive:Base  
{  
    public:  
        Derive(){ cout<<"Derive class"<<endl; }  
};  
  
void main(){  
    Derive d;  
}
```

- A. Base class
Derive class
- B. Derive class
Base class
- C. Base class
- D. Derive class

Câu 19. Kết quả chương trình là :

```
#include<iostream>
```



```

using namespace std;

class Base{
    protected:
        int xVal;
    public:
        Base(){ xVal = 5; }
    public:
        void Print(){
            cout<<"xVal = "<<xVal<<endl;
        }
};

class Derive:public Base
{
    public:
        Derive(int x){xVal = x;}
};

void main(){
    Derive d(10);
    d.Print();
}

```

- A. Màn hình xuất hiện: xVal = 5
- xVal = 10
- B. Màn hình xuất hiện: xVal = 5
- C. Màn hình xuất hiện: xVal = 10
- D. Chương trình bị lỗi.

Câu 20. Kết quả chương trình là :

```

#include<iostream>

using namespace std;

class Base{

```

```

protected:
    int xVal;

public: Base(){ xVal = 5; }

public: void Print(){
    cout<<"xVal = "<<xVal<<endl;
}
};

class Derive:protected Base
{
    public: Derive(int x){xVal = x;}
};

void main(){
    Derive d(10);
    d.Print();
}

```

A. Màn hình xuất hiện: xVal = 5

xVal = 10

B. Màn hình xuất hiện: xVal = 5

C. Màn hình xuất hiện: xVal = 10

D. Chương trình bị lỗi.

Câu 21. Kết quả chương trình là :

```
#include<iostream>
```

```
using namespace std;
```

```
class BaseA{
```

```
    protected:
```

```
        int A;
```

```
    public:
```

```
        BaseA(){ A = 5; }
```

```
        void Print(){ cout<<"A = "<<A<<endl; }
```

```

};

class BaseB{
    protected:
        int B;
    public:
        BaseB(){ B = 10; }
        void Print(){ cout<<"B = "<<B<<endl; }
};

class Derive:public BaseA,public BaseB{};

void main(){
    Derive d();
    d.BaseA::Print();
}

```

- A. Chương trình bị lỗi.
- B. Màn hình xuất hiện: A = 5
- C. Màn hình xuất hiện: B = 10
- D. Màn hình xuất hiện: A = 5
B = 10