

1 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;

class Exam
{
    int prop;
public:
    Exam();
    Exam(int);
    Exam(const Exam&);
    ostream& operator<<(ostream&, const Exam);
};
Exam::Exam() {

}
Exam::Exam(int prop)
    : prop(prop)
{

}
Exam::Exam(const Exam& obj)
{
    this->prop = obj.prop;
}
ostream& operator<<(ostream& o, const Exam obj) Friend
{
    o << obj.prop;
    return o;
}
int main()
{
    Exam obj1(1);
    Exam obj2 = obj1;
    cout << obj2;
    return 0;
}
```

A. Another Answer

B. Runtime Error

C. Compile Error //int Exam::prop is private

2 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;
```

```

class Base
{
    public:
        Base();
        Base(const Base &);
        void operator=(const Base &);
};
Base::Base()
{
    cout << "A";
}
Base::Base(const Base& obj)
{
    cout << "B";
}
void Base::operator=(const Base& obj)
{
    cout << "C";
}
int main()
{
    Base obj1;
    Base obj2;
    obj2 = obj1;
    system("Pause");
    return 0;
}

```

A. Another Answer

B. Runtime Error

C. Compile Error

D. ABC

E. AAB

F. AAC



3 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```

class Base
{
    double R, I;
    public:
        Base();
        Base operator()(double, double);
        Base operator()(double);
        friend ostream& operator<<(ostream&, const Base&);
};

```

```

Base::Base() : R(0), I(0)
{
}
Base Base::operator()(double R, double I)
{
    this->R += R;
    this->I += I;
    return *this;
}
Base Base::operator() (double R)
{
    this->R += R;
    return *this;
}
ostream& operator<<(ostream& o, const Base& obj)
{
    o <<"Base (" << obj.R << ", " << obj.I << ").";
    return o;
}
int main()
{
    Base obj1, obj2;
    obj2 = obj1(3.2, 5.3);
    obj1(6.5, 2.7);
    obj2(1.9);
    cout << obj1 << obj2;
    system("pause");
    return 0;
}

```

- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. Base (9.7, 8).Base (5.1, 5.3).**
- E. Base (4.7, 8).Base (2.1, 5.3).
- F. Base (9, 8).Base (5, 5).

5 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Exam
```

```
{
```

```
    public:
```

```
        operator string();
```

```
};
```

```
Exam::operator string()
```

```
{
```

```
    return "ITF";
```

```
}
```

```
int main()
```

```
{
```

```
    Exam obj;
```

```
    string str = obj;
```

```
    cout << str;
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. Garbage Value

E. ITF



6 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;

class Complex
{
    double R;
    double I;
public:
    Complex();
    Complex operator()(const double&, const double&);
    Complex operator()(const double&);
    void Func();
};

Complex::Complex()
    : R(0), I(0)
{

}

Complex Complex::operator()(const double& R, const double& I)
{
    this->R += R;
    this->I += I;
    return *this;
}

Complex Complex::operator()(const double& R)
{
    this->R += R;
    return *this;
}

void Complex::Func()
{
    cout << "(" << R << ", " << I << ")";
}

int main()
{
    Complex obj_x, obj_y;
    obj_y = obj_x(3.2, 5.3); //Y 3.2, 5.3 x 3.2, 5.3
    obj_x(6.5, 2.7); //x 9.7, 8
    obj_y(1.9); //y 5.1, 5.3
    obj_x.Func();
    obj_y.Func();
    return 0;
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. (9.7, 8)(5.1, 5.3)

E. (3.2, 5.3)(3.2, 5.3)

7 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
int Func(int, int);
```

```
int Func(int, int, int = 15);
```

```
int main()
```

```
{
```

```
    cout << Func(2, 4);
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

```
int Func(int x, int y)
```

```
{
```

```
    return x * y;
```

```
}
```

```
int Func(int x, int y, int z = 5)
```

```
{
```

```
    return x * y * z;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

//overload is ambiguous

D. 5

E. 40

F. 8

8 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Complex
```

```
{
```

```
    int R;
```

```
    int I;
```

```
    public:
```

```
        Complex(int, int);
```

```
        ~Complex();
```

```
        Complex operator+(const Complex&);
```

```
        friend ostream& operator<<(ostream&, const Complex&);
```

```

};
Complex::Complex(int R, int I)
    : R(R), I(I)
{

}
Complex::~~Complex()
{

}
Complex Complex::operator+(const Complex& obj)
{
    return Complex(this->R + obj.R, this->I + obj.I);
}
ostream& operator<<(ostream& o, const Complex& obj)
{
    o << obj.R << "+i" << obj.I;
    return o;
}
int main()
{
    Complex obj_x(1, 2), obj_y(3, 4);
    Complex obj_z = obj_x + obj_y; //4, 6
    cout << obj_z;
    return 0;
}

```

- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. 4+i6**
- E. 2+i2

9 Kết quả thực hiện đoạn chương trình sau là gì? (**Khác câu 7 ở đoạn bôi đen**)

```

#include<iostream>
using namespace std;

```

```

int Func(int, int);
int Func(int, int, int);

```

```

int main()
{
    cout << Func(2, 4);
    system("pause");
    return 0;
}
int Func(int x, int y)
{

```

```
        return x * y;
    }
    int Func(int x, int y, int z = 5)
    {
        return x * y * z;
    }
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 40

E. 5

F. 8

//ở trên nó khai báo Func(int, int, int) mà ko phải là Func(int, int, int = 5)

//nếu viết thành Func(int, int, int = 5) thì xảy r lỗi **ambiguous**

10 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    double x;
```

```
    public:
```

```
        Base(double = 0.0);
```

```
        operator double();
```

```
        operator int();
```

```
};
```

```
Base::Base(double x) : x(x)
```

```
{ }
```

```
Base::operator double()
```

```
{
```

```
    return this->x;
```

```
}
```

```
Base::operator int()
```

```
{
```

```
    return int(this->x);
```

```
}
```

```
int main()
```

```
{
```

```
    Base obj1(9.7), obj2(2.6);
```

```
    double x = double(obj1);
```

```
    int y = int(obj2);
```

```
    cout << x << y;
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 93

E. 9.72

F. 97.2



12 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;

using namespace std;

class BaseA
{
    public:
        BaseA& operator=(const BaseA&);
};
BaseA& BaseA::operator=(const BaseA& obj)
{
    cout << "AOBaseA.";
    return *this;
}
class BaseB
{
    BaseA arr[2];
};
int main()
{
    BaseB obj1, obj2;
    obj1 = obj2;
    system("pause");
    return 0;
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. AOBaseA.
- E. AOBaseA.AOBaseA.**



13 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;

class Complex
{
    double R, I;
public:
    Complex(double, double);
    Complex(double);
    friend Complex operator+(Complex, Complex);
    friend ostream& operator<<(ostream&, Complex);
};
Complex::Complex(double R, double I) : R(R), I(I)
{

}
Complex::Complex(double R)
{
    this->R = this->I = R;
}
Complex operator+(Complex obj1, Complex obj2)
{
    return Complex(obj1.R + obj2.R, obj1.I + obj2.I);
}
ostream & operator<<(ostream &o, Complex obj)
{
    o << obj.R << "," << obj.I;
    return o;
}
int main()
{
    Complex obj1(3, 4);
    Complex obj2 = obj1 + 6;
    cout << obj2;
    system("pause");
    return 0;
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. 8, 9
- E. 8, 4
- F. 9,10**

14 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
struct Struct
```

```
{
```

```
    public:
```

```
        int Addition(int, int = 10);
```

```
        double Addition(int, double);
```

```
};
```

```
int Struct::Addition(int x, int y)
```

```
{
```

```
    return (x * y + 2);
```

```
}
```

```
int main()
```

```
{
```

```
    Struct obj;
```

```
    cout << obj.Addition(1) << obj.Addition(3, 4);
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 1812

E. 314

F. 1212

G. 1218

16 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;

class ExamA {
public:
    virtual ~ExamA();
    void operator delete(void*);
};
ExamA::~~ExamA()
{

}
void ExamA::operator delete(void*)
{
    cout << "delete ExamA";
}
class ExamB : public ExamA
{
public:
    void operator delete(void*);
};
void ExamB::operator delete(void* ptr)
{
    cout << "delete ExamB";
}

int main()
{
    ExamA* ptr = new ExamB;
    delete ptr;
    return 0;
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. delete ExamBdelete ExamA
- E. delete ExamAdelete ExamB
- F. delete ExamA
- G. delete ExamB

18 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
ostream& operator<<(ostream& o, int var_x)
```

```
{
```

```
    return o;
```

```
}
```

```
int main()
```

```
{
```

```
    cout << 5;
```

```
    cin.get();
```

```
    return 0;
```

```
}
```

L i không s

d ng bi n

var_x

N u o <<

var_x => gây

ra l i quy

do trong th

vi n ã n p

toán t <<

A. Another Answer

B. Runtime Error

C. Compile Error //ambiguous for operator <<

D. 6

E. 5

19 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;

class Base
{
    int x;
    public:
        Base(int = 1);
        friend void operator<<(ostream &, Base);
};
Base::Base(int x) : x(x)
{}
void operator<<(ostream &o, Base obj)
{
    o << obj.x;
}

int main()
{
    Base obj1(2), obj2; //obj1 2, obj2 1
    cout << obj1 << obj2;
    system("pause");
    return 0;
}
```

A. Another Answer

B. Runtime Error

C. Compile Error //no match for 'operator<<' (operand types are 'void' and 'Base')

D. 21

E. 22

F. 11

23 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Exam
```

```
{
```

```
    public:
```

```
        int prop;
```

```
        Exam* operator->();
```

```
};
```

```
Exam* Exam::operator->()
```

```
{
```

```
    return this;
```

```
}
```

```
int main()
```

```
{
```

```
    Exam obj;
```

```
    obj->prop = 10;
```

```
    cout << obj.prop << obj->prop;
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 1010

E. 1111

30 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    int x;
```

```
    public:
```

```
        Base(int=1);
```

```
        friend ostream& operator<<(Base&, ostream &);
```

```
};
```

```
Base::Base(int x) : x(x)
```

```
{ }
```

```
ostream& operator<<(Base& obj, ostream &o)
```

```
{
```

```
    o << obj.x;
```

```
    return o;
```

```
}
```

```
int main()
```

```
{
```

```
    Base obj(2);
```

```
    obj << cout;
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 2

E. Garbage value

F. Another Answer

G. 1

32 Kết quả thực hiện đoạn chương trình sau là gì? (khác câu 23 ở chỗ sau)

```
#include<iostream>
using namespace std;

class Exam
{
    int prop;
public:
    Exam* operator->();
    ostream& operator<<(ostream& , const Exam&);
};
Exam* Exam::operator->()
{
    return this;
}
ostream& operator<<(ostream& o, const Exam& obj)
{
    o << obj.prop;
    return o;
}

int main()
{
    Exam obj;
    obj->prop = 10;
    cout << obj.prop << obj->prop;
    return 0;
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error //int Exam::prop private**
- D. 1111
- E. 1010
- F. Another Answer

33 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Exam
```

```
{
```

```
    public:
```

```
        int prop;
```

```
        Exam(int = 0);
```

```
        int& operator[](const int&);
```

```
        int operator[](const int&) const;
```

```
};
```

```
Exam::Exam(int prop)
```

```
    : prop(prop)
```

```
{}
```

```
int& Exam::operator[](const int& index)
```

```
{
```

```
    cout << "0";
```

```
    return this->prop;
```

```
}
```

```
int Exam::operator[](const int& index) const
```

```
{
```

```
    cout << "1";
```

```
    return this->prop;
```

```
}
```

```
void Func(const Exam& obj)
```

```
{
```

```
    int var_x = obj[2];
```

```
}
```

```
int main()
```

```
{
```

```
    Exam obj(7);
```

```
    obj[3] = 8;
```

```
    int var_x = obj[2];
```

```
    Func(obj);
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 110

E. 111

F. 001

G.011

37 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    int x;
```

```
    public:
```

```
        Base(int = 1);
```

```
        int operator,(int);
```

```
};
```

```
Base::Base(int x) : x(x)
```

```
{ }
```

```
int Base::operator,(int x)
```

```
{
```

```
    return this->x;
```

```
}
```

```
int main()
```

```
{
```

```
    Base obj;
```

```
    int x = 2;
```

```
    cout << obj.operator,(x);
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. Another Answer

E. 1

F. 2

38 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Point
```

```
{
```

```
    int x, y;
```

```
    public:
```

```
        Point();
```

```
        Point& operator()(int, int);
```

```
        friend ostream& operator<<(ostream&, const Point&);
```

```
};
```

```
Point::Point() : x(0), y(0)
```

```
{ }
```

```
Point& Point::operator()(int x, int y)
```

```
{
```

```
    this->x = x;
```

```
    this->y = y;
```

```
    return *this;
```

```
}
```

```
ostream& operator<<(ostream& o, const Point& obj)
```

```
{
```

```
    o << obj.x << obj.y;
```

```
    return o;
```

```
}
```

```
int main()
```

```
{
```

```
    Point obj;
```

```
    obj(3, 2);
```

```
    cout << obj;
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 23

E. 32

43 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    int x;
```

```
    public:
```

```
        Base(int = 1);
```

```
        int operator,(int);
```

```
        friend int operator, (Base&, int);
```

```
};
```

```
Base::Base(int x) : x(x)
```

```
{ }
```

```
int operator, (Base& o, int x)
```

```
{
```

```
    return x;
```

```
}
```

```
int main()
```

```
{
```

```
    Base obj;
```

```
    int x = 2;
```

```
    cout << (obj, x);
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error //ambiguous overload for 'operator,'

D. 1

E. 2

44 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include<iostream>
using namespace std;

class Base
{
    int x;
    public:
        Base(int);
        const Base operator+(const Base&) const;
        Base& operator+=(const Base&);
        friend ostream& operator<<(ostream&, const Base&);
};
Base::Base(int x) : x(x)
{}
const Base Base::operator+(const Base& obj) const
{
    return Base(this->x + obj.x);
}
Base& Base::operator+=(const Base& obj)
{
    this->x += obj.x;
    return *this;
}
ostream& operator<<(ostream& o, const Base& obj)
{
    o << obj.x;
    return o;
}

int main()
{
    int x = 1, y = 2, z = 3;
    z += x + y;
    Base obj1(x), obj2(y), obj3(z);
    obj3 += obj1 + obj2;
    cout << obj1 << obj2 << obj3;
    system("pause");
    return 0;
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 123

E. 129

F. 126

