1)

class Distance {

private:

int meter;

friend int addFive(Distance);

public:

Distance() : meter(2) {}

};

int addFive(Distance d) {

d.meter += 5;

return d.meter;

}

int main() {

Distance D;

cout << "Distance: " << addFive(D);

return 0;

}

Distance: 7

2)

class ClassB;

class ClassA {

public:

ClassA() : numA(3) {}

private:

int numA;

friend int add(ClassA, ClassB);

};

class ClassB {

public:

ClassB() : numB(1) {}

private:

int numB;

friend int add(ClassA, ClassB);

};

int add(ClassA objectA, ClassB objectB) {

return (objectA.numA + objectB.numB);

}

int main() {

ClassA objectA;

ClassB objectB;

cout << "Sum: " << add(objectA, objectB);

return 0;

}

Sum: 4

3)

class ClassB;

class ClassA {

private:

int numA;

friend class ClassB;

public:

ClassA() : numA(5) {}

};

class ClassB {

private:

int numB;

public:

ClassB() : numB(4) {}

int add() {

ClassA objectA;

return objectA.numA + numB;

}

};

int main() {

ClassB objectB;

cout << "Sum: " << objectB.add();

return 0;

}

Sum: 9

4)

template <class T>

class Number {

private:

T num;

public:

Number(T n) : num(n) {}

T getNum() {

return num\*3;

}

};

int main() {

Number<int> numberInt(2);

Number<double> numberDouble(2.2);

cout << "int Number = " << numberInt.getNum() << endl;

cout << "double Number = " << numberDouble.getNum() << endl;

return 0;

}

int Number = 6

double Number = 6.6

5)

template <class T, class U, class V = char>

class ClassTemplate {

private:

T var1;

U var2;

V var3;

public:

ClassTemplate(T v1, U v2, V v3) : var1(v1), var2(v2), var3(v3) {}

void printVar() {

cout << "var1 = " << var1 << endl;

cout << "var2 = " << var2 << endl;

cout << "var3 = " << var3 << endl;

}

};

int main() {

ClassTemplate<int, double> obj1(7, 7.7, 'c');

cout << "obj1 values: " << endl;

obj1.printVar();

ClassTemplate<double, char, bool> obj2(8.8, 'a', false);

cout << "\nobj2 values: " << endl;

obj2.printVar();

return 0;

}

obj1 values:

var1 = 7

var2 = 7.7

var3 = c

obj2 values:

var1 = 8.8

var2 = a

var3 = 0

6)

class Box

{

private:

int length;

public:

Box() : length(2) {}

friend int printLength(Box);

};

int printLength(Box b)

{

b.length += 10;

return b.length;

}

int main()

{

Box b;

cout << " Length of box : " << printLength(b) << endl;

return 0;

}

Length of box : 12

7)

class B; //forward declaration.

class A{

int x;

public:

void setdata(int i) {

x = i;

}

friend void max(A, B);

};

class B{

int y;

public:

void setdata(int i) {

y = i;

}

friend void max(A, B);

};

void max(A a, B b){

if (a.x >= b.y)

cout << a.x << std::endl;

else

cout << b.y << std::endl;

}

int main(){

A a;

B b;

a.setdata(9);

b.setdata(3);

max(a, b);

return 0;

}

9

8)

class space

{

int x;

int y;

int z;

public:

void setdata(int a, int b, int c);

void display(void);

friend void operator- (space &s);

};

void space::setdata(int a, int b, int c)

{

x = a; y = b; z = c;

}

void space::display(void)

{

cout << x << " " << y << " " << z << "\n";

}

void operator- (space &s)

{

s.x = -s.x;

s.y = -s.y;

s.z = -s.z;

}

int main()

{

space s;

s.setdata(5, 2, 9);

cout << "s:";

s.display();

-s;

cout << "-s:";

s.display();

return 0;

}

s:5 2 9

-s:-5 -2 -9

9) Bunu çalıştırın sonucu yorumlayın niye?

class Box

{

int capacity;

public:

Box(int cap){

capacity = cap;

}

friend void show();

};

void Box::show()

{

Box b(10);

cout << "Value of capacity is: " << b.capacity << endl;

}

int main(int argc, char const \*argv[])

{

show();

return 0;

}

show’dan dolayı çalımadı

10) çalıştırın Niye?

class B

{

int b;

public:

B(int i){

b = i;

}

};

class C

{

B b;

public:

C(int i){

b = B(i);

}

friend void show();

};

void show()

{

C c(10);

cout << "value of b is: " << c.b.b << endl;

}

int main(int argc, char const \*argv[])

{

show();

return 0;

}

int b private

11)

class B

{

int b;

public:

B(){}

B(int i){

b = i;

}

int show(){

return b;

}

};

class C

{

B b;

public:

C(int i){

b = B(i);

}

friend void show();

};

void show()

{

C c(8);

cout << "value of b is: " << c.b.show() << endl;

}

int main(int argc, char const \*argv[])

{

show();

return 0;

}

value of b is: 8

12) Çalıştırın Niye, açıklayın

class B

{

int b;

public:

B(){}

B(int i){

b = i;

}

int show(){

return b;

}

};

class C

{

B b;

public:

C(int i){

b = B(i);

}

friend void show(){

C c(10);

cout << "value of b is: " << c.b.show() << endl;

}

};

int main(int argc, char const \*argv[])

{

show();

return 0;

}

13) Niye

class B

{

int b;

public:

B(){}

B(int i){

b = i;

}

int show(){

return b;

}

};

class C

{

B b;

public:

C(int i){

b = B(i);

}

friend void show(){

C c(10);

cout << "value of b is: " << c.b.show() << endl;

}

};

int main(int argc, char const \*argv[])

{

C c(1);

c.show();

return 0;

}

14)

class A

{

int x = 50;

friend class B; // friend class.

};

class B

{

public:

void display(A &a)

{

cout << "value of x is : " << a.x;

}

};

int main()

{

A a;

B b;

b.display(a);

return 0;

}

value of x is : 50

15)

template <class T>

class Test

{

private:

T val;

public:

static int count;

Test() { count++; }

};

template<class T>

int Test<T>::count = 1;

int main()

{

Test<int> a;

Test<int> b;

Test<double> c;

cout << Test<int>::count << endl;

cout << Test<double>::count << endl;

return 0;

}

3

2

16) char is 1 byte, size of int : 4 bytes,

template<class T, class U>

class A {

T x;

U y;

static int count;

};

int main() {

A<char, char> a;

A<int, int> b;

cout << sizeof(a) << endl;

cout << sizeof(b) << endl;

return 0;

}

2

8

17)

template <class T, int max>

int arrMin(T arr[], int n)

{

int m = max;

for (int i = 0; i < n; i++)

if (arr[i] < m)

m = arr[i];

return m;

}

int main()

{

int arr1[] = { 10, 20, 15, 12 };

int n1 = sizeof(arr1) / sizeof(arr1[0]);

char arr2[] = { 1, 2, 3 };

int n2 = sizeof(arr2) / sizeof(arr2[0]);

cout << arrMin<int, 10000>(arr1, n1) << endl;

cout << arrMin<char, 256>(arr2, n2);

return 0;

}

10

1

18)

template<class T, class U, class V = double>

class A

{

T x;

U y;

V z;

};

int main()

{

A<int, int> a;

A<double, double> b;

cout << sizeof(a) << endl;

cout << sizeof(b) << endl;

return 0;

}

16

24

19)template <class T>

T sum(T num1, T num2)

{

T C = num1 + num2;

return C;

}

int main()

{

int A = 5;

int B = 10;

double X = 3.5;

double Y = 4.6;

cout << sum(A, B) << " " << sum(X, Y) << endl;

return 0;

}

15 8.1

20)

template <class T>

int sum(T num1, T num2)

{

T C = num1 + num2;

return C;

}

int main()

{

int A = 5;

int B = 10;

double X = 3.5;

double Y = 4.6;

cout << sum(A, B) << " " << sum(X, Y) << endl;

return 0;

}

15 8

21)

template <class X, class Y>

Y sum(X num1, Y num2)

{

Y C = num1 + num2;

return C;

}

int main()

{

int A = 5;

int B = 10;

double X = 3.5;

double Y = 4.6;

cout << sum(A, Y) << endl;

return 0;

}

9.6

22)

template <class T>

class Sample {

T a;

T b;

public:

Sample(T a, T b)

{

this->a = a;

this->b = b;

}

void print()

{

cout << a << " " << b << endl;

}

};

int main()

{

Sample<int> S(10, 20);

S.print();

return 0;

}

10 20