#1 Что выведет программа?

#include <iostream>

#include <memory>

class A

{

public:

A() { func(); }

virtual void func() { std::cout << "A" ; }

};

class B : public A

{

public:

B() { func(); }

void func() { std::cout << "B" ; }

};

int main()

{

std::auto\_ptr< A > obj( new B );

obj->func();

return 0;

}

ABB +

BBB

BBA

AAA

Ошибка выполнения

Ошибка компиляции

**Объяснение**: В конструкторе класса виртуальная таблица методов еще не создана, поэтому виртуальные методы вызываются как обычные, то есть вызывается последний реализованный для этого класса метод.

#2 Что выведет cout?

#include <iostream>

int & calculate(int &x)

{

x++;

return x;

}

int main() {

int x, y = 10;

x = ++calculate(y);

std::cout << x << " " << y;

return 0;

}

11 12

12 12 +

11 11

12 11

**Объяснение**: 12 12

#3 Какое значение будет выведено на экран в результате выполнения следующего кода:

#define SQUARE(x) ((x)\*(x))

int m = 4;

int n = SQUARE(++m);

std::cout << n;

16

25

36

Неопределённое поведение +

**Объяснение**: При выполнении директивы #define выполняется текстовая замена каждого встретившегося в теле функции символа X на (++m). Таким образом имеем (++m)\*(++m), а это неопределённое поведение

#4 Что будет выведено на экран?

#include <iostream>

#include <stdlib.h>

using namespace std;

namespace A

{

void func(int n)

{

cout << n << endl;

}

};

namespace A

{

void func(float d)

{

cout << d << endl;

}

};

namespace B

{

void func(char c)

{

cout << c << endl;

}

};

using namespace A;

using namespace B;

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

{

func( 'Z');

return 0;

}

90

Z +

ошибка компиляции

нет правильного ответа

#5 Дан класс:

class c1 {

int i;

public:

int get\_i() { return i; }

void put\_i(int j) { i = j; }

};

Являются ли функции get\_i() и put\_i () встраиваемыми функциями?

только put\_i ()

только get\_i()

да +

нет

#6 Что выведет cout?

#include <iostream>

using namespace std;

struct A {

A();

A(int v) { cout << v; }

};

int main()

{

A a(0), a2, a3(2);

}

20

ошибка выполнения

02

ошибка линковки +

не скомпилируется

**Объяснение**: Пропущена реализация конструктора по умолчанию.

#7 Каков результат работы следующей программы?

#include<iostream>

using namespace std;

main() {

enum { india, is = 7, GREAT };

cout << india << " " << GREAT;

}

0 1

0 2

0 8 +

Ошибка компиляции

#8 Чему будет равно значение p после выполнения следующей строки кода:

int p = 1 > 2 ? (3 > 3 ? 1 : 2) : (3 < 3 ? 3 : 4);

2

3

1

4 +

#9 Что произойдёт при попытке откомпилировать и запустить код?

#include <iostream>

using namespace std;

int preinc(int &x) {

return ++x;

}

int postinc(int &x) {

return x++;

}

int main(void) {

int x = 5;

cout << preinc(x);

cout << postinc(x);

cout << x;

return 0;

}

на экран будет выведено 667 +

на экран будет выведено 567

на экран будет выведено 677

на экран будет выведено 555

код не скомпилируется

программа проявит неопределенное поведение (undefined behavior) во время работы

#10 Какое свойство ООП дает концепцию повторного использования?

Абстракция

Инкапсуляция

Наследование +

Ни один из вышеперечисленных.

#11 Что выведет программа:

#include <iostream>

class A

{

public:

A() { std::cout << "A()"; }

A(const A& a){ std::cout << "copyA()"; }

A& operator=(const A& a)

{

std::cout << "=()";

}

};

int main()

{

A a1;

A a2 = a1;

}

A()copyA()=()

A()A()=()

A()A()copyA()=()

A()copyA() +

A()=()

**Объяснение**: Программа выведет: A()copyA()

Copy constructor is called when a new object is created from an existing object, as a copy of the existing object. And assignment operator is called when an already initialized object is assigned a new value from another existing object.

a2 = a1; // calls assignment operator, same as "a2.operator=(a1);" A a3 = a1; // calls copy constructor, same as "A a3(a1);"

#12 В одной из записей функции main имеются два аргумента. Какая минимальная величина может быть у аргумента argc?

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

0

1 +

2

3

4

**Объяснение**: Минимальная величина аргумента argc будет равна 1, поскольку в качестве первого аргумента всегда передаётся имя программы.

#13 Что выведет на экран следующая программа?

#include <iostream>

void qwer(int &param)

{

param += 15;

}

int main(void)

{

int a = 14;

void (\*realFunction)(int &) = &qwer;

realFunction(a);

std::cout << a;

return 0;

}

undefined behavior

15

29 +

программа не скомпилируется

14

#14 Что будет выведено на экран:

#include <iostream>

using namespace std;

void func(class A\* pA);

class A

{

public:

A()

{func(this);}

virtual void function()=0;

};

class B : public A

{

public:

virtual void function()

{

cout<<"function B"<<endl;

}

};

void func(A\* pA)

{

pA->function();

}

int main()

{

A\*a = new B;

}

Ошибка времени выполнения +

function B

Ничего

Ошибка компиляции

**Объяснение**: Ошибка времени выполнения - вызов чисто виртуальной функции. Т.к. первым вызывается конструктор базового класса, то попытка вызова в нем чисто виртуальной функции, через обертку в виде стороней функции приведет к ошибке времени выполнения.

#15 Что выведет следующая программа:

#include <iostream>

#include <cstdio>

int main() {

std::cout << printf("boom!");

return 0;

}

printf("boom!")

Ошибка компиляции (несоответствие типов)

Ошибка компиляции (библиотека stdio входит только в стандартную библиотеку языка С)

Нет правильного ответа +

boom!

**Объяснение**: должно быть выведено "boom!5", т.к. printf() возвращает int, равный количеству символов, выведенных в stdout (<http://www.cplusplus.com/reference/clibrary/cstdio/printf/> ). std::cout по умолчанию синхронизирован с stdout, поэтому вначале в stdout попадет "boom!", а затем - "5" (<http://www.cplusplus.com/reference/iostream/ios_base/sync_with_stdio/> ).

Задача 1.

Ввести с клавиатуры координаты начала и конца отрезка на плоскости. Ввести координаты точки. Проверить, лежит ли эта точка на отрезке.

#include <cstdlib>

using namespace std;

double max(double x, double y)

{

if (x < y) {

return x;

}

return x;

}

double min(double x, double y)

{

if (x > y) {

return y;

}

return x;

}

bool thc(double x, double y, double z, double w, double a, double b)

{

double k, c;

if (z == x) {

return (a == x && b >= min(y, w) && x <= max(y, w));

}

k = (w - y) / (z - x);

c = y - k \* x;

return b == a \* k + c;

}

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

{

setlocale(LC\_ALL, "Russian");

double x, y; // Координаты начала отрезка

double z, w; // Координаты конца отрезка

double a, b; // Координаты точки

bool result;

cout << "Координаты начала отрезка: ";

cin >> x >> y;

cout << "Координаты конца отрезка: ";

cin >> z >> w;

cout << "Координаты точки: ";

cin >> a >> b;

result = thc(x, y, z, w, a, b);

cout << result << endl;

system("pause");

return 0;

}

thc(x, y, z, w, a, b);

cout << result << endl; system("pause");

return 0;

}

Задача 2.

Заполнить массив с клавиатуры. Найти его наименьший положительный элемент и его индекс. Вывести результат на экран.

#include <iostream>

#include <iostream>

using namespace std;

const int ARR\_SIZE = 5;

int main()

{

setlocale(LC\_CTYPE, "Russian");

int i, X[ARR\_SIZE];

cout << "Заполните массив." << endl;

for (i = 0; i < ARR\_SIZE; i++) {

cout << "Введите значение № " << i + 1 << ": ";

cin >> X[i];

}

int Xmin = -1, min = -1;

for (i = 0; i < ARR\_SIZE; i++) {

if (X[i] > 0) {

if (min < 0) {

Xmin = X[i];

min = i;

}

else {

if (X[i] < Xmin) {

Xmin = X[i];

min = i;

}

}

}

}

if (min < 0) {

cout << "Наим. полож. элемент отсутствует" << endl;

}

else {

cout << "Наим. полож. элемент: " << X[min] << endl

<< "Его номер: " << min + 1 << endl;

}

return 0;

}

Задача 3.

Найти минимальное и максимальное значений из трех чисел

#include <iostream>

using namespace std;

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

{

setlocale(LC\_ALL, "");

int a, b, c;

cout << "Введите 3 числа через пробел: ";

cin >> a >> b >> c;

int min = a;

int max = a;

// Нахождениа минимума

if (b < min) {

min = b;

}

if (c < min) {

min = c;

}

// Нахождение максимума

if (b > max) {

max = b;

}

if (c > max) {

max = c;

}

cout << "Max: " << max << endl;

cout << "Min: " << min << endl;

return 0;}

Задача 4. Повышенная сложность.

Построить таблицу всех различных разбиений заданного натурального числа N на сумму трех натуральных слагаемых (разбиения, отличающиеся порядком слагаемых, различными не считаются).

#include <iostream>

int main()

{

std::cout << "Enter a natural number: ";

unsigned int N;

std::cin >> N;

std::cout << "The number " << N << " can be present by next three summands:\n";

unsigned int limit\_1\_summand = (N / 3) + 1;

unsigned int limit\_2\_summand;

for (unsigned int i = 0; i < limit\_1\_summand; i++) {

limit\_2\_summand = ((N - i) / 2) + 1;

for (unsigned int j = i; j < limit\_2\_summand; j++) {

std::cout << i << " " << j << " " << N - i - j << std::endl;

}

}

std::cin.get();

std::cin.get();

return 0;

}

Задача 5.

Ввести целочисленный массив из N 'элементов с клавиатуры. Отсортировать его по возрастанию методом пузырька.

#include <iostream>

using namespace std;

int main()

{

int \*arr; // указатель для выделения памяти под массив

int size; // размер массива

// Ввод количества элементов массива

cout << "n = ";

cin >> size;

if (size <= 0) {

// Размер масива должен быть положитлеьным

cerr << "Invalid size" << endl;

return 1;

}

arr = new int[size]; // выделение памяти под массив

// заполнение массива

for (int i = 0; i < size; i++) {

cout << "arr[" << i << "] = ";

cin >> arr[i];

}

int temp; // временная переменная для обмена элементов местами

// Сортировка массива пузырьком

for (int i = 0; i < size - 1; i++) {

for (int j = 0; j < size - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

// меняем элементы местами

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

// Вывод отсортированного массива на экран

for (int i = 0; i < size; i++) {

cout << arr[i] << " ";

}

cout << endl;

delete [] arr; // освобождение памяти;

return 0;

}

Задача 6.

Ввести целочисленный массив из N элементов с клавиатуры. Найти максимальный элемент этого массива.

#include <iostream>

using namespace std;

int main()

{

int \*arr; // указатель для выделения памяти под массив

int size; // размер массива

// Ввод количества элементов массива

cout << "n = ";

cin >> size;

if (size <= 0) {

// Размер масива должен быть положитлеьным

cerr << "Invalid size" << endl;

return 1;

}

arr = new int[size]; // выделение памяти под массив

// заполнение массива

for (int i = 0; i < size; i++) {

cout << "arr[" << i << "] = ";

cin >> arr[i];

}

// Нахождение максимального элемента

int max = arr[0];

for (int i = 1; i < size; i++) {

if (arr[i] > max) {

max = arr[i];

}

}

// Вывод результата на экран

cout << "max = " << max << endl;

delete [] arr; // освобождение памяти

return 0;

}