**Тест по курсу C++**

***Теоретические вопросы***

1. Каким ключевым словом обозначаются встраиваемые функции?

Ваш ответ:

inline

1. Каким будет значение переменной c после выполнения следующего кода?

int a, b = a = 0, c;

if (b != a++)

c = 1;

else if (b == --a)

c = 2;

Варианты ответа:

* 0
* 1
* 2
* код не скомпилируется.

Ваш ответ:

1. Пусть а = -5. Сколько раз выполнится тело цикла

while (a<0) a++;

Варианты ответа:

* 0
* 4
* 5
* 6

Ваш ответ:

1. Какие ключевые слова входят в состав инструкции "если":

Выберите возможные варианты:

* Else (с маленькой буквы “else”)
* then
* if
* elseif (если отдельно 2 слова else if)
* unless

Ваш ответ:

1. Что из перечисленного является объявлением указателя в С++:

* int\* a;
* int &a;
* int a&;
* int\* &a;

Ваш ответ:

1. Дана функция:

int sum(int a, int b)

{

return a + b;

}

Как правильно вызвать эту функцию?

Выберите возможные варианты:

* int sum(int a = 7, int b = 8);
* sum(int 7, int 8);
* sum(7, 8);
* sum() : 7, 8;
* sum(int a = 7, int b = 8);

Ваш ответ:

1. Выберите правильные утверждения о конструкторе класса:

* Конструктор объявляется в точности так же, как и обычный метод класса.
* Конструктор не возвращает значения.
* Конструктор может иметь любое имя.
* Конструктор имеет то же имя, что и класс.
* Имя конструктора начинается с символа ~.

Ваш ответ:

1. Содержит ли этот код ошибки?

class A {

static void Foo();

};

void A::Foo() { }

Варианты ответа:

* да
* нет

Ваш ответ: код скомпилируется, но за пределами класса мы не сможешь обратиться к методу Foo, если он не public.

1. Какие виды комментариев есть в C++?

Выберите возможные варианты:

* # comment
* // comment
* ; comment
* <!-- comment -->
* /\* comment \*/

Ваш ответ:

1. Что означает конструкция

throw()

в объявлении функции? (например void f() throw();)

Варианты ответа:

* Такое объявление указывает, что функция не должна генерировать исключения.
* Такое объявление указывает, что функция может сгенерировать любое исключение.
* Такое объявление не корректно, произойдет ошибка времени компиляции.
* Такое объявление указывает, что функция может сгенерировать исключения только из стандартной библиотеки или же исключения такого же типа как и параметры функции.

Ваш ответ:

1. В каком порядке инициализируются поля в классе?

Варианты ответа:

* Порядок инициализации не гарантируется
* В порядке перечисления инициализаторов в списке инициализации конструктора
* В порядке их объявления

Ваш ответ:

1. Что верно о следующем коде:

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

{

int a[3] = { 1, 2, 3 };

int b[2] = { 1, 2 };

a = b;

return 0;

}

Варианты ответа:

* Код скомпилируется и успешно выполнится
* Код скомпилируется, но возникнет ошибка времени выполнения
* Код не скомпилируется

Ваш ответ:

1. Сколько раз выполнится цикл:

for (int i = 0; i <= 5; i += 3);

Варианты ответа:

* 5 раз,
* 6 раз
* это бесконечный цикл,
* ни одного раза,
* 2 раза,
* 3 раза

Ваш ответ:

1. Как правильно объявить массив?

Варианты ответа:

* int array[];
* int array[5];
* int array[] = new array[5];
* int array[] = new int[];

Ваш ответ:

1. Что произойдет, если объявить следующие функции:

int Square (int width, int length = 3);

int Square (int size);

и вызвать функцию

int s = Square (10);

Варианты ответа:

* Код скомпилируется и успешно выполнится
* Код скомпилируется, но возникнет ошибка времени выполнения
* Код не скомпилируется

Ваш ответ:

1. Какой будет вывод в консоль и почему?

#include <iostream>   
#include <string>   
   
void print(int v)   
{   
    std::cout << "int:" << v << std::endl;   
}   
   
void print(bool v)   
{   
    std::cout << "bool:" << v << std::endl;   
}   
   
void print(std::string v)   
{   
    std::cout << "std::string:" << v << std::endl;   
}   
   
int main()   
{   
    print(1);   
    print(true);   
    print("Hello world");   
}

Ваш ответ:

int:1

bool:1

bool:1

Согласно стандарту, выбор одной из перегруженных функций осуществляется по правилу «Наиболее подходящей функции» (англ. «Best viable function»). <code class=\"language-cpp\">\"Hello world\"</code> имеет тип <code class=\"language-cpp\">const char\*</code> и приведение его к POD-типу <code class=\"language-cpp\">bool</code> целесообразней чем к типу <code class=\"language-cpp\">std::string</code>.

1. Для каких целей применяется ключевое слово const?

Ваш ответ: Ключевое const слово указывает, что значение переменной является константой и сообщает компилятору, чтобы предотвратить его изменение

1. Как защитить объект от копирования?

Ваш ответ:

* Удалить оператор присвоения operator=()=delete
* перенести конструктора в private
* **private**:

// copy and assignment prohibited

NonCopyable( **const** NonCopyable& );

**void** NonCopyable::operator=( **const** NonCopyable& );

***Практические вопросы***

1. Создайте класс “A”, инкапсулирующий динамический массив. Введите в класс необходимые на Ваш взгляд данные и методы, чтобы приведенный ниже код выполнялся:

{

A a1;

A a2(10); //10 – размер массива

A a3 = a2;

a1 = a3;

a2 = A(20);

const A a4(5);

for(int i=0; i<a2.size(); i++)

{

cout<<a4[i];

}

}

class A {

private:

int n;

int\* arr;

public:

A() {

n = 0;

arr = {};

}

A(int size) {

n = size;

arr = new int[size];

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

arr[i] = 0;

}

}

A(const A& \_arr) {

n = \_arr.n;

arr = new int[n];

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

arr[i] = \_arr.arr[i];

}

int size()const {

int count = 0;

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

count++;

}

return count;

}

~A() {

if (n > 0)

delete[] arr;

}

A& operator=(const A& ar) {

if (n > 0)

delete[] arr;

n = ar.n;

arr = new int[n];

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

arr[i] = ar.arr[i];

return \*this;

}

const int& operator[] (const int& index)const {

return arr[index];

}

};

2. Добавьте в программу Eng\_float модуля 12 в класс Distance перегруженную операцию присваивания.

Distance& operator= (float m)

{

float fltfeet = MTF \* m;

feet = int(fltfeet);

inches = 12 \* (fltfeet - feet);

return \*this;

}

Distance& operator=(const Distance& d)

{

feet = d.feet;

inches = d.inches;

return \*this;

}

3. Дан класс Point:

class Point{

int x,y;

…

};

Обеспечьте выполнение:

{

Point pt1(1,1) pt2(2,2), pt3;

pt3 = pt1 + pt2;

pt2 +=pt1;

pt3 = pt1 + 5;

…

}

class Point {

int x, y;

public:

Point() : x(0), y(0) {};

Point(int xx, int yy) : x(xx), y(yy) {};

Point& operator + (const Point& point) const

{

Point p(this->x + point.x, this->y + point.y);

return p;

}

Point& operator += (const Point& point)

{

x += point.x,

y += point.y;

return \*this;

}

Point& operator + (int i) const

{

Point p(this->x + i, this->y + i);

return p;

}

void show()

{

cout << x << y << endl;

}

};

4. Напишите класс, который реализует функционал стека. Класс **Stack** должен иметь:

* Открытый целочисленный фиксированный массив длиной 10.
* Открытое целочисленное значение для отслеживания длины стека.
* Открытый метод с именем reset(), который будет сбрасывать длину и все значения элементов на 0.
* Открытый метод с именем push(), который будет добавлять значение в стек. Метод push() должен возвращать значение false, если массив уже заполнен, в противном случае — true.
* Открытый метод с именем pop() для возврата значений из стека. Если в стеке нет значений, то должно выводиться предупреждающее сообщение.
* Открытый метод с именем print(), который будет выводить все значения стека.

Следующий код функции main():

| int main()  {  Stack stack;  stack.reset();  stack.print();  stack.push(3);  stack.push(7);  stack.push(5);  stack.print();  stack.pop();  stack.print();  stack.pop();  stack.pop();  stack.print();  return 0;  } |
| --- |

Должен производить следующий результат:

( )  
( 3 7 5 )  
( 3 7 )  
( )

class Stack {

public:

int array[10] {};

int sL{};

int aL = sizeof(array) / sizeof(\*array);

void reset()

{

sL = 0;

for (int n : array)

{

n = 0;

}

}

bool push(int n)

{

if (sL < aL)

{

array[sL] = n;

sL++;

return true;

}

return false;

}

void pop()

{

if (sL == 0) {

cout << "Stack пуст";

}

else

{

array[sL] = 0;

sL--;

}

}

void print()

{

cout << "( ";

if (sL != 0)

{

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

{

cout << array[i] << " ";

}

}

cout << ")" << endl;

}

};