

Тест по курсу C++

Теоретические вопросы

1. Каким ключевым словом обозначаются встраиваемые функции?

Ваш ответ: **inline**

2. 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**

3. Пусть `a = -5`. Сколько раз выполнится тело цикла `while (a<0) a++;`

Варианты ответа:

- 0
- 4
- **5**
- 6

Ваш ответ: **3**

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

Выберите возможные варианты:

- **else**
- then
- **if**
- elseif
- unless

Ваш ответ: **1, 3**

5. Что из перечисленного является объявлением указателя в C++:

- **int* a;**
- `int &a;`
- `int a&;`

- `int* &a;`

Ваш ответ: **1**

6. 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: **3, 5**

7. 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: **2, 4**

8. Содержит ли этот код ошибки?

```
class A {
    static void Foo();
};
void A::Foo() { }
```

Варианты ответа:

- да
- **нет**

Ваш ответ: **2**

9. Какие виды комментариев есть в C++?

Выберите возможные варианты:

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

Ваш ответ: 2, 5

10. 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

11. 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: 2

12. 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: 3

13. 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: 5

14. Как правильно объявить массив?

Варианты ответа:

- `int array[];`
- `int array[5];`
- `int array[] = new array[5];`
- `int array[] = new int[];`

Ваш ответ: 2

15. 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: 3

16. 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, bool, bool**

17. Для каких целей применяется ключевое слово const?

Ваш ответ: **для неизменяемой переменной**

18. Как защитить объект от копирования?

Ваш ответ: **перегрузка оператора присваивания и удаление конструктора копирования.**

Практические вопросы

1. Создайте класс “А”, 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 {  
public:  
    A() {  
        s = 0;  
        arr = {};  
    }  
    A(int size) {  
        s = size;  
        arr = new int[size];  
        for (int i = 0; i < size; i++)  
        {  
            arr[i] = 0;  
        }  
    }  
}
```

```

A(const A& _arr) {
    s = _arr.s;
    arr = new int[s];
    for (int i = 0; i < s; i++)
    {
        arr[i] = _arr.arr[i];
    }
}

int size()const {
    int count = 0;
    for (int i = 0; i < s; i++)
    {
        count++;
    }
    return count;
}

A& operator=(const A& ar) {
    if (s > 0)
        delete[] arr;
    s = ar.s;
    arr = new int[s];
    for (int i = 0; i < s; i++)
        arr[i] = ar.arr[i];
    return *this;
}

const int& operator[] (const int& index)const {
    return arr[index];
}

private:
    int s;
    int* arr;
};

```

2. Добавьте в программу Eng_float модуля 12 в класс Distance перегруженную операцию присваивания.

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
{
public:
    Point() {}
    Point(int xx, int yy)

```

```

{
    x = xx;
    y = yy;
}
Point operator+(const Point& pt)
{
    Point res(this->x + pt.x, this->y + pt.y);
    return res;
}
void operator+=(const Point& pt)
{
    this->x += pt.x;
    this->y += pt.y;
}
Point operator+(const int& n)
{
    Point res(this->x + n, this->y + n);
    return res;
}
int x, y;

};

```

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:
    void reset()
    {
        size = 0;
        for (int i : arr)
        {
            i = 0;
        }
    }
    bool push(int num)
    {
        if (size < 10)
        {
            arr[size] = num;
            size++;
            return true;
        }
        return false;
    }
    int pop()
    {
        int res = arr[size - 1];
        arr[size - 1] = 0;
        size--;
        return res;
    }
    void print()
    {
        std::cout << "(" << " ";
        for (int i = 0; i < size; i++)
        {
            std::cout << arr[i] << " ";
        }
        std::cout << ")" << " " << std::endl;
    }
    int arr[10];
    int size = 0;
};

```