

1

Какво ще изведе следният код?

```
struct MyStruct {  
    char c;  
    int i;  
};  
MyStruct s;  
std::cout << sizeof(s);
```

- a) 5
- б) 8
- в) 9
- г) 12

Отговор: б) 8 - Заради padding-a, структурата ще има размер 8 байта.

2

Какво трябва да се попълни на празното място, за да се запълнят празнините (padding) и да се минимизира размера на структурата?

```
struct MyStruct {  
    char c;  
    -----;  
    int i;  
};
```

- а) short s
- б) double d
- в) char pad[3]
- г) float f

Отговор: в) `char pad[3]` - За запълване на 3-те байта `padding`.

3

Каква е разликата между структура и клас в C++?

- а) Няма разлика
- б) Структурите по подразбиране имат публични членове, а класовете - частни
- в) Класовете имат функции-членове, а структурите - не
- г) Структурите могат да наследяват други структури, но не и класове

Отговор: б) Структурите по подразбиране имат публични членове, а класовете – частни – Това е основната разлика между структура и клас.

4

Какво ще изведе следният код?

```
class MyClass {  
public:  
    int x;  
    MyClass(int val) : x(val) {}  
};  
MyClass obj(10);
```

`std::cout << obj.x;`

- а) Нищо, ще има грешка
- б) 0
- в) 10
- г) Неопределена стойност

Отговор: в) 10 - Конструкторът инициализира x със стойността 10.

5

Кое от следните твърдения за конструкторите е вярно?

- а) Конструкторът не може да бъде претоварен
- б) Конструкторът може да бъде наследен
- в) Конструкторът може да бъде претоварен
- г) Конструкторът винаги трябва да има параметри

Отговор: в) Конструкторът може да бъде претоварен - Могат да съществуват няколко конструктора с различни параметри.

6

Каква е грешката в следния код?

```
class MyClass {  
private:  
    int x;  
public:  
    void setX(int val) { x = val; }  
    int getX() { return x; }  
};  
MyClass obj;  
obj.setX(5);  
std::cout << obj.x;
```

- а) Няма грешка
- б) x не може да бъде достъпен директно, защото е private
- в) Методите setX и getX трябва да бъдат private
- г) Липсва конструктор за инициализация

Отговор: б) `x` не може да бъде достъпен директно, защото е `private` - Трябва да се използва `getX()` за достъп.

7

Какво ще изведе следният код?

```
class MyClass {  
public:  
    MyClass() { std::cout << "Constructor"; }  
    ~MyClass() { std::cout << "Destructor"; }  
};  
int main() {  
    MyClass obj;  
    return 0;  
}
```

- a) Constructor
- б) Destructor
- в) ConstructorDestructor
- г) Constructor Destructor

Отговор: в) ConstructorDestructor - Първо се извиква конструкторът, след това деструкторът.

8

Коя е правилната сигнатура на оператора за присвояване?

- а) `ClassName& operator=(ClassName& other)`
- б) `ClassName operator=(const ClassName& other)`
- в) `ClassName& operator=(const ClassName& other)`
- г) `ClassName& operator=(const ClassName other)`

Отговор: в) `ClassName& operator=(const ClassName& other)` - Това е сигнатурата на копиращия оператор за присвояване.

9

Коя е правилната сигнатура на move конструктора?

- а) `ClassName(ClassName& other)`
- б) `ClassName(const ClassName& other)`
- в) `ClassName(ClassName&& other)`
- г) `ClassName(const ClassName&& other)`

Отговор: в) `ClassName(className&& other)` - Това е стандартната сигнатура на move конструктор.

10

Какво трябва да се попълни на празното място, за да стане кодът валиден?

```
class MyClass {  
public:  
    MyClass& operator=(MyClass&& other) noexcept {  
        if (this != &other) {  
            -----;  
        }  
        return *this;  
    }  
private:  
    AnotherClass x;  
};
```

a) delete other

б) x = std::move(other.x)

в) std::swap(*this, other)

г) std::swap(&other, &x)

Отговор: б) `x = std::move(other.x)` – За да се използва move семантика.

11

Какво ще изведе следният код?

```
class MyClass {  
public:  
    int* data;  
    MyClass(int size) {  
        data = new int[size];  
    }  
    ~MyClass() {  
        delete[] data;  
    }  
};  
int main() {  
    MyClass obj1(10);  
    MyClass obj2 = std::move(obj1);  
    std::cout << obj1.data;  
    return 0;  
}
```

- а) Адрес на data
- б) 0
- в) Undefined behavior
- г) Грешка при компилация

Отговор: в) Undefined behavior - След move, obj1.data сочи към изтритата памет.

12

Как се отваря файл за двоично писане с `std::ofstream`?

- а) `std::ofstream file("filename");`
- б) `std::ofstream file("filename", std::ios::app);`
- в) `std::ofstream file("filename", std::ios::binary);`
- г) `std::ofstream file("filename", std::ios::in);`

Отговор: в) `std::ofstream file("filename", std::ios::binary);` - Флагът `std::ios::binary` отваря файла в двоичен режим.

13

Какво ще изведе следният код?

```
std::ofstream file("example.txt");  
file << "Hello, World!";  
file.close();  
std::ifstream fileRead("example.txt");  
std::string str;  
fileRead >> str;  
std::cout << str;
```

- a) Hello, World!
- б) Hello
- в) World!
- г) Нищо

Отговор: б) Hello - >> операторът чете до първото бяло пространство.

14

Каква е грешката в следния код?

```
std::ofstream file("example.txt", std::ios::binary);  
int num = 100;  
file.write(reinterpret_cast<char*>(&num), sizeof(num));  
file.close();  
std::ifstream fileRead("example.txt");  
fileRead >> num;  
std::cout << num;
```

- а) Няма грешка
- б) Четене на двоични данни като текст е неправилно
- в) Проблем с отварянето на файла
- г) Проблем с cast-a

Отговор: б) Четене на двоични данни като текст е неправилно – Двоични данни трябва да се четат със read.

15

Какво е приятелска функция?

- а) Функция, която има достъп до `private` и `protected` членове на клас
- б) Функция, която може да бъде извикана само от вътрешността на класа
- в) Функция, която не може да бъде извикана извън класа
- г) Функция, която наследява класа

Отговор: а) Функция, която има достъп до `private` и `protected` членове на клас - Декларира се с ключовата дума `friend`.

16

Какво ще изведе следният код?

```
class MyClass {  
    int x;  
public:  
    MyClass(int val) : x(val) {}  
    friend int getX(const MyClass& obj);  
};  
int getX(const MyClass& obj) {  
    return obj.x;  
}  
int main() {  
    MyClass obj(10);  
    std::cout << getX(obj);  
    return 0;  
}
```

- а) Нищо, ще има грешка
- б) 0
- в) 10
- г) Неопределена стойност

Отговор: в) 10 - Приятелската функция `getX` има достъп до `private` члена `x`.

17

Какво трябва да се попълни на празното място, за да стане кодът валиден?

```
class MyClass {  
public:  
    int x;  
    MyClass(int val) : x(val) {}  
    bool operator==(const MyClass& other) const {  
        return _____;  
    }  
};
```

- a) `this->x == other.x`
- б) `this.x == other.x`
- в) `this->x = other.x`
- г) `this.x = other.x`

Отговор: а) `this->x == other.x` - Сравнява стойностите на `x`.

18

Как се декларира шаблонна функция?

а) `template <class T> T functionName(T param);`

б) `template <typename T> T functionName(T param);`

в) `template <type T> T functionName(T param);`

г) `template <T> T functionName(T param);`

Отговор: а) и б) са двата възможни варианта

19

Какво ще изведе следният код?

```
class Integer{
public:
    Integer(int _data) : data(_data) {};
private:
    int data;
}

template <typename T>
T add(T a, T b) {
    return a + b;
}

int main() {
    Integer a(5), b(3);
    std::cout << add(a, b);
    return 0;
}
```

- а) 0
- б) 8
- в) Грешка
- г) 3

Отговор: в) Грешка - за класа Integer няма дефиниран оператор +.

20

Как се декларира статичен член на клас?

- a) `static int member;`
- б) `int static member;`
- в) `int member static;`
- г) `member static int;`

Отговор: a) `static int member;`

21

Каква е грешката в следния код?

```
class MyClass {  
public:  
    static int x;  
    MyClass(int val) {  
        x = val;  
    }  
};  
int main() {  
    MyClass obj1(10);  
    MyClass obj2(20);  
    std::cout << obj1.x << " " << obj2.x;  
    return 0;  
}
```

- а) Няма грешка
- б) Статичният член трябва да бъде инициализиран извън класа
- в) x трябва да бъде non-static
- г) Конструкторът трябва да бъде static

Отговор: б) Статичният член трябва да бъде инициализиран извън класа - Инициализация: `int MyClass::x = 0;`

22

Какво ще изведе следният код?

```
class Base {  
public:  
    virtual void show() { std::cout << "Base"; }  
};  
class Derived : public Base {  
public:  
    void show() override { std::cout << "Derived"; }  
};  
int main() {  
    Base* b = new Derived();  
    b->show();  
    delete b;  
    return 0;  
}
```

- a) Base
- б) Derived
- в) BaseDerived
- г) Грешка при компилация

Отговор: б) Derived – Полиморфизъм, извиква се методът на Derived.

23

Каква е грешката в следния код?

```
class Base {  
public:  
    virtual void show() = 0;  
};  
int main() {  
    Base b;  
    b.show();  
    return 0;  
}
```

- а) Няма грешка
- б) Абстрактен клас не може да бъде инстанциран
- в) Методът show трябва да бъде non-virtual
- г) Методът show не трябва да бъде чисто виртуален

Отговор: б) Абстрактен клас не може да бъде инстанциран - Base е абстрактен клас и не може да се създават обекти от него.

24

Какво ще изведе следният код?

```
class A {  
public:  
    virtual void show() { std::cout << "A"; }  
};  
class B {  
public:  
    virtual void show() { std::cout << "B"; }  
};  
class C : public A, public B {  
public:  
    void show() override { std::cout << "C"; }  
};  
int main() {  
    C obj;  
    A* a = &obj;  
    B* b = &obj;  
    a->show();  
    b->show();  
    return 0;  
}
```

- a) AC
- б) BC
- в) CC
- г) AB

Отговор: в) СС - през указателите се извиква виртуалния метод `show` за обект от тип `C`

25

Какво ще изведе следният код?

```
class MyClass {  
public:  
    int x;  
    MyClass(int val) : x(val) {}  
    MyClass operator+(const MyClass& other) {  
        return MyClass(this->x + other.x);  
    }  
};  
int main() {  
    MyClass obj1(5), obj2(3);  
    MyClass obj3 = obj1 + obj2;  
    std::cout << obj3.x;  
    return 0;  
}
```

- a) 5
- б) 3
- в) 8
- г) Грешка при компилация

Отговор: в) 8 – Операторът + събира x на двата обекта.

26

Какво ще изведе следният код?

```
class Base {  
public:  
    virtual void show() { std::cout << "Base"; }  
};  
class Derived : public Base {  
public:  
    void show() override { std::cout << "Derived"; }  
};  
int main() {  
    Base b;  
    Derived d;  
    Base* bp = &b;  
    bp->show();  
    bp = &d;  
    bp->show();  
    return 0;  
}
```

- a) BaseBase
- б) BaseDerived
- в) DerivedDerived
- г) Грешка при компиляция

Отговор: б) BaseDerived – Първо се извиква show на Base, после show на Derived чрез полиморфизъм.

27

Какво ще изведе следният код?

```
class MyClass {  
public:  
    static int x;  
    MyClass() { x++; }  
};  
int MyClass::x = 0;  
int main() {  
    MyClass obj1, obj2, obj3;  
    std::cout << MyClass::x;  
    return 0;  
}
```

- a) 0
- б) 1
- в) 2
- г) 3

Отговор: г) 3 – Статичният член x се увеличава при всяка инстанция.