

[Табло](#) / [Моите курсове](#) / [Бакалаври, летен семестър 2020/2021](#) / [И](#)
/ [Обектно-ориентирано програмиране \(И, КН1\), летен семестър 2020/2021](#) / Блиц тестове
/ [Блиц тест 2 | класове \(увод и жизнен цикъл\)](#)

Започнат на	четвъртък, 6 май 2021, 14:30
Състояние	Завършен
Приключен на	четвъртък, 6 май 2021, 14:40
Изминало време	9 мин. 54 сек.
Точки	17,00/17,00
Оценка	10,00 от 10,00 (100%)

Въпрос 1

Правилен
отговор1,00 от
максимално
1,00 точки

За всяка от член-променливите посочете нейната видимост.

```
class A {  
    int a1;  
private:  
    int a2;  
    int a3;  
public:  
    int a4;  
private:  
};  
  
struct B {  
    int b1;  
private:  
    int b2;  
    int b3;  
public:  
    int b4;  
private:  
};
```

A::a1	private	✓
A::a2	private	✓
A::a3	private	✓
A::a4	public	✓
B::b1	public	✓
B::b2	private	✓
B::b3	private	✓
B::b4	public	✓

Your answer is correct.



Въпрос 2

Правилен
отговор1,00 от
максимално
1,00 точки

Нека са дадени следните дефиниции:

```
class Test {  
};  
  
void by_ref(Test& param) {}  
void by_cref(const Test& param) {}  
void by_value(Test param) {}  
void by_cvalue(const Test param) {}  
  
int main()  
{  
    by_ref(Test());  
    by_cref(Test());  
    by_value(Test());  
    by_cvalue(Test());  
}
```

Във функцията `main` се съдържат поредица от обръщения към функциите, при които им подаваме обект от тип `Test`. За всяко от тях посочете дали е коректно или ще предизвика грешка.

<code>by_ref(Test());</code>	грешка	✓
<code>by_cref(Test());</code>	коректно	✓
<code>by_value(Test());</code>	коректно	✓
<code>by_cvalue(Test());</code>	коректно	✓

Your answer is correct.

Въпрос 3

Правилен
отговор1,00 от
максимално
1,00 точки

Кое от следните е коректният начин да укажем, че функцията `foo` приема константна референция (const reference) към обект от тип `Bar`?

- ☐ `void foo(const Bar& ref)`
- ☐ `void foo(Bar const& ref)`
- ☒ И двете са коректни



Your answer is correct.



Въпрос 4

Правилен
отговор1,00 от
максимално
1,00 точки

Посочете вярното твърдение за дадения по-долу код:

```
#include <iostream>

class Test {
    int& ref;
public:
    Test(int& ref) : ref(ref)
    {}

    void increment() const
    {
        ++ref;
    }
};

int main()
{
    int var = 10;
    Test obj(var);
    obj.increment();
    std::cout << var << "\n";
}
```

- ☐ a. Кодът ще се компилира без грешки. Програмата извежда 10.
- ☐ b. Ще се получи грешка при изпълнение на програмата (runtime error) -- инкрементиране **ref** в **const** функцията **increment**.
- ☒ c. Кодът ще се компилира без грешки. Програмата извежда 11. ✓
- ☐ d. Ще се получи грешка при компилиране (compiler error) -- инкрементиране **ref** в **const** функцията **increment**.

Your answer is correct.



Въпрос 5

Правилен
отговор1,00 от
максимално
1,00 точки

Какъв вид копиране извършва автоматично генерираният оператор за присвояване в дадения по-долу фрагмент?

```
class Internal {
    int a;
    Internal& operator=(const Internal& other)
    {
        a = other.a;
        return *this;
    }
};

class External {
    Internal obj;
    int b;
};

int main()
{
    External x, y;
    x = y;
}
```

- ☐ Задължително е да се използва покомпонентно копиране с оператора за присвояване на всеки от членовете.
- ☐ Компиляторът може да използва тривиално копиране (например с `memcpy` или `memmove`)
- ☒ Компиляторът не може да генерира оператор за присвояване. ✓

Your answer is correct.

Операторът за присвояване на `Internal` е `private`.

Въпрос 6

Правилен
отговор1,00 от
максимално
1,00 точки

Какъв вид копиране извършва автоматично генерираният оператор за присвояване в дадения по-долу фрагмент?

```
class Internal {
    int a;
public:
    Internal& operator=(const Internal& other)
    {
        a = other.a;
        return *this;
    }
};

class External {
    Internal obj;
    int b;
};

int main()
{
    External x, y;
    x = y;
}
```

- ☒ Задължително е да се използва покомпонентно копиране с оператора за присвояване на всеки от членовете. ✓
- ☐ Компиляторът може да използва тривиално копиране (например с `memcpy` или `memmove`)
- ☐ Компиляторът не може да генерира оператор за присвояване.

Your answer is correct.



Въпрос 7

Правилен
отговор1,00 от
максимално
1,00 точки

Какъв вид копиране извършва автоматично генерираният оператор за присвояване в дадения по-долу фрагмент?

```
class Test {  
    int var = 0;  
};  
  
int main()  
{  
    Test a, b;  
    a = b;  
    return 0;  
}
```

- ☐ Задължително е да се използва покомпонентно копиране с оператора за присвояване на всеки от членовете.
- ☒ Компиляторът може да използва тривиално копиране (например с `memcpy` или `memmove`)
- ☐ Компиляторът не може да генерира оператор за присвояване.



Your answer is correct.

Въпрос 8

Правилен
отговор1,00 от
максимално
1,00 точки

Какъв вид копиране извършва автоматично генерираният оператор за присвояване в дадения по-долу фрагмент?

```
class Internal {  
    int a;  
};  
  
class External {  
    Internal obj;  
    int b;  
};  
  
int main()  
{  
    External x, y;  
    x = y;  
}
```

- ☐ Задължително е да се използва покомпонентно копиране с оператора за присвояване на всеки от членовете.
- ☐ Компиляторът не може да генерира оператор за присвояване.
- ☒ Компиляторът може да използва тривиално копиране (например с `memcpy` или `memmove`)



Your answer is correct.



Въпрос 9

Правилен
отговор1,00 от
максимално
1,00 точки

Коя форма на оператора за присвояване дефинира даденият по-долу код?

```
class Test {  
    int var = 0;  
public:  
    Test& operator=(Test other)  
    {  
        std::swap(var, other.var);  
        return *this;  
    }  
};
```

- ☐ a. канонична форма
- ☐ b. return *this форма
- ☒ c. copy-and-swap
- ☐ d. никоя от посочените



Your answer is correct.

Въпрос 10

Правилен
отговор1,00 от
максимално
1,00 точки

Каква версия на оператора за присвояване ще генерира компилаторът за дадения по-долу фрагмент?

```
class Test {  
    int var = 0;  
};  
  
int main()  
{  
    Test a, b;  
    a = b;  
    return 0;  
}
```

- ☐ Test& operator=(Test)
- ☐ Test& operator=(Test &)
- ☐ Компилаторът няма да генерира оператор за присвояване, защото променливата var е private.
- ☒ Test& operator=(const Test &)



Your answer is correct.



Въпрос 11

Правилен
отговор1,00 от
максимално
1,00 точки

Каква версия на оператора за присвояване ще генерира компилаторът за дадения по-долу фрагмент?

```
class Test {  
    int var = 0;  
    const int cvar = 0;  
};  
  
int main()  
{  
    Test a, b;  
    a = b;  
}
```

- ☐ `Test& operator=(const Test &)`
- ☐ `Test& operator=(Test)`
- ☒ Компилаторът не може да генерира оператор за присвояване, защото класът съдържа константа. ✓
- ☐ `Test& operator=(Test &)`

Your answer is correct.

Въпрос 12

Правилен
отговор1,00 от
максимално
1,00 точки

Посочете вярното твърдение за дадения по-долу код:

```
class Bar {  
    int var = 0;  
public:  
    Bar& operator=(const Bar& other)  
    {  
        var = other.var;  
        return *this;  
    }  
};  
  
void foo(Bar obj)  
{  
    obj = obj;  
}  
  
int main()  
{  
    Bar obj;  
    foo(obj);  
}
```

- ☐ a. В `Bar::operator=` има грешка. Не можем да достъпим `other.var`, защото тази променлива е `private` за обекта `other`.
- ☐ b. В `Bar::operator=` има проблем, компилаторът ще даде грешка, защото не правим проверка за `this != &other`.
- ☐ c. В `main` има грешка. Не можем да създадем обекта `obj`, защото в `Bar` не сме дефинирали default конструктор, а такъв няма да се генерира автоматично, защото в класа има дефиниран `operator=`.
- ☐ d. В `main` има грешка. Не можем да подадем `obj` на `foo`, защото в класа няма копиращ конструктор.
- ☒ e. Кодът ще се компилира без грешки ✓

Your answer is correct.



Въпрос 13

Правилен
отговор1,00 от
максимално
1,00 точкиЩе даде ли грешка компилаторът, ако дефинираме оператора за присвояване на `Test` така:

```
class Test {  
    int var;  
public:  
    bool operator=(Test& other)  
    {  
        if (this == &other)  
            return false;  
        var = other.var;  
        return true;  
    }  
};
```

Изберете едно:

- ☐ Истина
- ☒ Лъжа ✓

Въпрос 14

Правилен
отговор1,00 от
максимално
1,00 точкиБезопасно ли е да копираме данни между два обекта, от произволен клас, с функцията `memmove`?

Изберете едно:

- ☐ Истина
- ☒ Лъжа ✓

Въпрос 15

Правилен
отговор1,00 от
максимално
1,00 точкиКоректно ли е един клас `Test` да има конструктор от тип `Test(Test other)`?

Изберете едно:

- ☐ Истина
- ☒ Лъжа ✓

Въпрос 16

Правилен
отговор1,00 от
максимално
1,00 точки

Възможно ли е в един клас да се дефинират няколко различни версии на оператора за присвояване?

Изберете едно:

- ☒ Истина ✓
- ☐ Лъжа

Въпрос 17

Правилен
отговор1,00 от
максимално
1,00 точки

Възможно ли е в един клас да се дефинират няколко различни копиращи конструктора?

Изберете едно:

- ☒ Истина ✓
- ☐ Лъжа

[◀ Блиц тест 1 | структури и обединения](#)

Отиди на ...

[Блиц тест 3 | оператори ▶](#)