

1. Разгледайте представеният фрагмент от код и определете какви са проблемите в него.

А) ред 6—неправилно е дефинирана константна препратка към обект от класа Foo;

Б) ред 7—променливата `f` е предадена като константна препратка и поради това нейната стойност не може да се променя;

В) ред 7—член-променливата `f.x_` е недостъпна и не може да се използва;

Г) всичко е наред;

```

1 class Foo {
2     friend int bar(const Foo& f);
3     int x_;
4     //...
5 };
6 int bar(const Foo& f) {
7     return 1+f.x_;
8 }

```

2. Дадена е дефиниция на класа X. В тялото на метода `f` на ред 5 трябва да се използва член-променливата `v_` на обекта, предаден като параметър `X* x`. Кой е правилният начин да се направи това?

А) `x.v_;`

Б) `(*x)->v_;`

В) член-променливата `v_` е скрита (**private**) и не може да се използва;

Г) `x->v_;`

```

1 class X {
2     int v_;
3 public:
4     void f(X* x) {
5         //...
6     };
7 };

```

3. Разгледайте представения фрагмент. Какво е дефинирано на ред 6?

А) променлива `bar` от тип `foo` за която се вика конструктор с параметър от типа `int` и стойност 0;

Б) временен обект от типа `foo`;

В) конструкцията е невалидна в C++;

Г) функция `bar` с един аргумент от тип `int` и тип на резултата `foo`;

```

1 class foo{
2 public:
3     foo(int v);
4 };
5 //...
6 foo bar(int);

```

4. Дадена е декларацията на класа X. Указателят, дефиниран в ред 7 — `X* px` — е насочен към динамично създаден масив от 10 елемента от типа X. Кой е правилният начин да се освободи динамичната памет, към която сочи `px`?

А) `delete px[ ];`

Б) `delete [ ] px;`

В) `delete px;`

Г) `free(px);`

```

1 class X{
2 public:
3     X(void);
4     //...
5 };
6 //...
7 X* px;

```

5. Кой от член-променливите `x_` и `y_` на класа B са достъпни в методите на класа D?

А) `y_;`

Б) `x_;`

В) нито една от двете член променливи ;

Г) `x_` и `y_;`

```

1 class B {
2     int x_;
3 protected:
4     int y_;
5     //...
6 };
7 class D: public B { /*...*/ };

```

6. Кой от методите `dump()` ще се извика в ред 12 — `B::dump()` или `D::dump()`?

А) `D::dump()`, тъй като изборът между двата метода се извършва динамично, а действителният тип, към който сочи променливата `pb` е `class D`;

Б) `B::dump()`, тъй като изборът между двата метода се извършва статично;

В) `B::dump()`, тъй като изборът между двата метода се извършва динамично, а действителният тип, към който сочи променливата `pb` е `class B`;

Г) `D::dump()`, тъй като изборът между двата метода се извършва статично;

```

1 class B {
2 public:
3     void dump(void) const
4     { /* ... */ };
5 };
6 class D: public B {
7 public:
8     void dump(void) const
9     { /* ... */ };
10 };
11 B* pb=new D;
12 pb->dump();

```

7. Разгледайте представения код. Каква операция се изпълнява в ред 7?

А) копиращ конструктор;

Б) конструкцията е невалидна в `C++`;

В) оператор за присвояване;

Г) конструктор по подразбиране, а след него — оператор за присвояване;

```

1 class X {
2 public:
3     X(void);
4     //...
5 };
6 X x1;
7 X x2=x1;

```

8. Разгледайте представения код. Каква операция се изпълнява в ред 7?

А) копиращ конструктор;

Б) конструктор по подразбиране;

В) конструкцията е невалидна в `C++`;

Г) в ред 7 има грешка, тъй като не е дефиниран копиращ конструктор за класа `X`;

```

1 class X {
2 public:
3     X(void);
4     //...
5 };
6 X x1;
7 X x2(x1);

```

9. Какво ще стане, ако методът `init_B()` в ред 10 генерира изключение?

А) изпълнението на конструктора ще се прекрати, ще се извика деструкторът на `A` за член-променливата `a_`, инициализирана в ред 9 и няма да бъде създаден обект от класа `B`;

Б) синтактична грешка в ред 9 — не е дефиниран типът на резултата;

В) изпълнението на конструктора ще се прекрати и обектът от класа `B` ще остане в невалидно състояние;

Г) ще се извика `std::terminate()`, тъй като в конструкторите не може да се генерират изключения;

```

1 class A {
2 public:
3     A(int n);
4     //...
5 };
6 class B {
7     A a_;
8 public:
9     B(int n) : a_(n) {
10         init_B();
11     }
12     //...
13 };

```

10. В кои случаи се изпълнява кодът между ред 7 и ред 9?

- А) винаги, когато в кода между ред 1 и 3 се генерира изключение;
- Б) когато в кода между ред 1 и 3 се генерира изключение, което е от класа `std::exception` или е наследник на класа `std::exception`;
- В) когато в кода между ред 1 и 3 се генерира изключение, което не е от класа `std::exception` и не е наследник на класа `std::exception`;
- Г) в ред 7 има синтактична грешка;

```

1 try {
2     //...
3 }
4 catch(const std::exception& ex) {
5     //...
6 }
7 catch(...) {
8     //...
9 }
```

11. Разгледайте представения фрагмент от код. Какви типове могат да се предават като фактически аргументи на шаблона в ред 8?

- А) само класове;
- Б) само класове и структури;
- В) само примитивни типове;
- Г) всякакви типове;

```

1 template<class T>
2 class stack {
3     T* data_;
4 public:
5     const T& top(void) const;
6     //...
7 };
8 stack<WHAT> st;
```

12. Разгледайте представения фрагмент от код. Приемете, че за класа `D` не е дефиниран оператор `operator<<(ostream&, const D&)`.

Определете какви са проблемите в представения код.

- А) тъй като класът `D` няма оператор за изход, то създаването на обект от класа `B<D>` в ред 8 и използването на метода `print()` в ред 9 е неправилно;
- Б) тъй като класът `D` няма оператор за изход, дефиницията на метода `print()` в ред 5 е неправилна;
- В) конструкцията в ред 8 е неправилна;
- Г) в ред 1 използването на ключовата дума `class` е неправилно — вместо нея трябва да се използва ключовата дума `typename`;

```

1 template<class T>
2 class B {
3     T data_;
4 public:
5     void print(void){cout << data_;}
6 };
7 class D { /*...*/};
8 B<D> var;
9 var.print();
```

1. Г	4. Б	7. А	10. В
2. Г	5. А	8. А	11. Г
3. Г	6. Б	9. А	12. А