

1. Разгледайте представения фрагмент от код и определете какви грешки има в него.

- А) всичко е наред;
- Б) ред 5—след като променливата е декларирана, тя не може да се дефинира в същият файл;
- В) ред 4—не може една променлива да се декларира два пъти;
- Г) ред 3—не може една и съща променлива да се дефинира два пъти;

```

1 int bar;
2 extern int foo;
3 int bar;
4 extern int foo;
5 int foo;

```

2. Разгледайте представения фрагмент от код и определете какви са проблемите в него.

- А) всичко е наред;
- Б) масивите v1 и v2 са дефинирани неправилно в ред 1;
- В) резултатът от операцията в ред 3 е недефиниран;
- Г) резултатът от операцията в ред 5 е недефиниран;

```

1 int v1[10], v2[10];
2 int s1=&v1[8]-&v1[3];
3 int s2=&v1[5]-&v1[2];
4 int* p1=&v1[0]+2;
5 int* p2=&v2[2]-3;

```

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

- А) ред 6—променливата f е предадена като константна препратка и поради това нейната стойност не може да се променя;
- Б) ред 5—неправилно е дефинирана константна препратка към обект от класа Foo;
- В) всичко е наред;
- Г) ред 6—член-променливата f.x_ е недостъпна и не може да се използва;

```

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

```

4. Разгледайте представения код и открийте грешките в него. Преценете кой от проблемите в кода ще се прояви първи?

- А) при дефиницията на метода на ред 4 не е указан типът на резултата;
- Б) при изпълнението на ред 7 конструкторът е недостъпен;
- В) всичко в представения код е наред;
- Г) при изпълнението на ред 8 член-променливата x_ е недостъпна;

```

1 struct X{
2     int x_;
3     // ...
4     X(void) {x_=0;}
5 };
6 // ...
7 X v;
8 v.x_=10;

```

5. Разгледайте представения фрагмент от код. Определете какви са проблемите в него?

- А) всичко е наред;
- Б) конструкцията в ред 5 е невалидна в C++;
- В) в ред 8 не може да се променя стойността на size_;
- Г) за метода, дефиниран на ред 4, не е определен типът на резултата;

```

1 class X {
2     const int size_;
3 public:
4     X(void)
5         : size_(128)
6     {}
7     void foo(void){
8         size_+=128;
9     }
10 };

```

6. Дадена е декларацията на класа X. Кой е правилният начин да се създаде динамичен обект от типа X, за който да се извика конструкторът на X с параметър 10?

- А) X* x=new X[10];
- Б) X* x=new X(10);
- В) X* x=new(X,10);
- Г) X* x=malloc(X,10);

```

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

```

7. В какъв ред се извикват деструкторите при унищожаване на обект от класа D?

- А) деструкторът на производния клас, деструкторът на базовия клас;
 Б) в реда, описан от програмиста;
 В) зависи от вида наследяване;
 Г) деструкторът на базовия клас, деструкторът на производния клас;

```

1 class B {
2     /*...*/
3 };
4 class D: public B {
5     /*...*/
6 };

```

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

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

```

1 class B {
2     public:
3         virtual 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();

```

9. Дадена е част от декларацията на класа X. Коя е правилната декларация на копиращ конструктор за класа X?

- А) `X(const X* x)`;
 Б) `void X(const X& x)`;
 В) `X(const X& x)`;
 Г) `X& X(const X& x)`;

```

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

```

10. Определете какви са проблемите в показания фрагмент от код?

- А) всичко е наред;
 Б) използването на `throw` в редове 2 и 8 е некоректно;
 В) използването на `throw` в ред 2 е некоректно;
 Г) не е допустимо да има два последователни `catch`-блока;

```

1 try {
2     throw;
3 }
4 catch(error& ex) {
5     //...
6 }
7 catch(...) {
8     throw;
9 }

```

11. Каква функция ще се инстанциира в ред 7?

- А) грешка — компилаторът не може сам да изведе аргументите на шаблона;
 Б) `fun<int>()`;
 В) `fun<double>()`;
 Г) грешка — променливите `r` и `v` трябва да се от един и същ тип;

```

1 template<class T>
2 T& fun(const T& val){
3     //...
4 }
5 double r;
6 int v;
7 r=fun(v);

```

12. Определете какви са проблемите в представеният фрагмент от код.

- А) няма проблеми;
 Б) не е извикан методът `close()` за обекта `from`, поради което файлът остава отворен;
 В) не е извикан методът `open()` за обекта `from`, поради което не може да се чете от файла;
 Г) в ред 2 не е указан режимът, в който трябва да се отвори файла;

```

1 void dump() {
2     ifstream from("temp.txt");
3     char ch;
4     while(from.get(ch))
5         cout.put(ch);
6 }
```

13. Кой от методите на потока показва, че следващата операция с потока няма да се изпълни успешно?

- А) `good()`; Б) `bad()`; В) `eof()`; Г) `fail()`;

14. Докога ще се изпълнява цикълът `while(cin>>i);`?

- А) докато потокът `cin` не премине в състояние `ios_base::badbit`;
 Б) цикълът е безкраен;
 В) изразът е некоректен, защото няма преобазуване от `istream` към `bool`;
 Г) докато потокът `cin` не премине в състояние `ios_base::failbit`;

15. Кой от изброените изрази гарантира, че след използването му потокът `istream in` ще бъде в състояние `ios_base::goodbit`?

- А) `in.setstate(~ios_base::failbit);` Б) `in.clear();`
 В) `in.setstate(ios_base::goodbit);` Г) `in.clear(~ios_base::badbit);`

16. Какво прави функцията `foo()` дефинирана в ред 1?

- А) копира съдържанието на файла "aa" върху файла "bb";
 Б) добавя съдържанието на файла "aa" в края на "bb";
 В) грешка в ред 2 и 3 – не е указан режимът, в който трябва да се отворят съответните файлове;
 Г) грешка в дефиницията на функцията – не са извикани методите `close()` на обектите `from` и `to`;

```

1 void foo() {
2     ifstream from("aa");
3     ofstream to("bb");
4     char ch;
5     while(from.get(ch))
6         to.put(ch);
7 }
```

17. Определете какви са проблемите в оператора за изход, дефиниран в ред 7.

- А) параметърът `out` от типа `ostream`, който се предава в ред 7, трябва да бъде `const`;
 Б) няма проблеми;
 В) методите `get_x()` и `get_y()`, дефинирани в ред 3 и 4, не трябва да бъдат `const`;
 Г) параметърът `p` от типа `point`, който се предава в ред 8, не трябва да бъде `const`;

```

1 class point {
2 public:
3     double get_x() const;
4     double get_y() const;
5     //...
6 };
7 ostream& operator<<(ostream& out,
8     const point& p) {
9     out<<' ('<<p.get_x()<<'_';
10    <<p.get_y()<<' )';
11    return out;
12 }
```

1. Г	6. Б	11. Б	16. А
2. Г	7. А	12. А	17. Б
3. Г	8. Б	13. Г	
4. В	9. В	14. Г	
5. В	10. В	15. Б	