

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

- А) ред 1—неправилно е използвана ключовата дума **enum**;
- Б) ред 6—не е коректно като резултат от функция да се връща стойността на локална променлива;
- В) ред 5—променливата `my_var` не може приема целочислени стойности;
- Г) ред 2—една функция не може да има тип на резултата `MY_TYPE`;

```
1 enum MY_TYPE {ONE, TWO, THREE};  
2 MY_TYPE fun() {  
3     MY_TYPE my_var;  
4     my_var=ONE;  
5     my_var=2;  
6     return my_var;  
7 }
```

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

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

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

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

- А) ред 2—не може да се дефинира масив от указатели **void***;
- Б) ред 3—не може да се дефинира променлива от типа **void****;
- В) няма проблеми;
- Г) ред 1—типа на аргументите на една функция не може да бъде **void**;

```
1 void fun(void);  
2 void* varray[20];  
3 void** ptr;  
4 void fun(void* arg);
```

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

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

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

5. Разгледайте представения фрагмент от код. Каква ще бъде стойността на променливата `w` след изпълнението на ред 7?

- А) грешка в ред 3 – не може да се връща препратка (reference) като резултат от функция;
- Б) `w=10`;
- В) `w=0`;
- Г) грешка в ред 2 – стойността на `c` не може да се променя в тялото на функцията;

```
1 int inc(const int& c){  
2     c+=10;  
3     return c;  
4 };  
5 //...  
6 int v=0;  
7 int w=inc(v);
```

6. Разгледайте представения фрагмент от код. Какви ще бъдат стойностите на променливите *i* и *j* след изпълнението на ред 6?

- A) *i*=0; *j*=19;
- B) *i*=1; *j*=20;
- B) *i*=2; *j*=21;
- Г) *i*=1; *j*=1;

```
1 void inc(int& v, int& w){
2     v+=1; w+=1;
3 }
4 //...
5 int i=1; int j=20;
6 inc(i, j);
```

7. Каква ще бъде стойността на променливата *j*, дефинирана в ред 1, след изпълнението на ред 8?

- A) *j*=2;
- B) *j*=42;
- B) стойността е недефинирана, тъй като променливата *j*, дефинирана в ред 1, не е инициализирана;
- Г) *j*=24;

```
1 int j;
2 int main(void) {
3     j+=1;
4     {
5         int j=20;
6         ::j+=22;
7     }
8     j+=1;
9     return 0;
10 }
```

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

- A) ред 2 – не може препратка (reference) да се инициализира с 0;
- B) всичко е наред;
- B) ред 4 – не може да се дефинира препратка (reference) без да се инициализира;
- Г) ред 3 – не може препратка (reference) да се инициализира с друг препратка;

```
1 int val=0;
2 int& v1=val;
3 int& v2=v1;
4 int& v3;
5 int* p=&val;
6 int& v4=*p;
```

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

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

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

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

- A) ред 7—променливата *f* е предадена като константна препратка и поради това нейната стойност не може да се променя;
- B) ред 6—неправилно е дефинирана константна препратка към обект от класа *Foo*;
- B) всичко е наред;
- Г) ред 7—член-променливата *f.x_* е недостъпна и не може да се използва;

```
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 }
```

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

- А) ред 5—променливата `f` е предадена като константна препратка и поради това нейната стойност не може да се променя;
- Б) ред 5—член-променливата `f.x_` е недостъпна и не може да се използва;
- В) ред 4—неправилно е дефинирана константна препратка към обект от класа `Foo`;
- Г) всичко е наред;

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

12. Разгледайте представеният фрагмент от код. Определете какви ще бъдат стойностите на член-променливите `f1.x_` и `f2.x_` след изпълнението на ред 11.

- А) `f1.x_=4`, `f2.x_=2`;
- Б) `f1.x_=2`, `f2.x_=4`;
- В) грешка в програмата—член-променливата `x_` е скрита и не може да бъде използвана в тялото на функцията `bar` в ред 6;
- Г) грешка в програмата—за член-функцията, дефинирана в ред 4, не е дефиниран типа на резултата;

```
1 class Foo {
2     int x_;
3 public:
4     Foo(void) :x_(2) {}
5     void bar(Foo& f) {
6         f.x_+=x_;
7     }
8 };
9 void f(void) {
10    Foo f1,f2;
11    f1.bar(f2);
12 }
```

13. Дадена е декларацията на класа `X`. В ред 7 трябва да се извика методът `f` на обекта `x`, обявен на ред 6. Кой е правилният начин да се направи това?

- А) `x.f`;
- Б) `x.f()`;
- В) `(*x).f()`;
- Г) `x->f()`;

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

14. 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) член-променливата `v_` е скрита (**private**) и не може да се използва;
- Б) `(*x)->v_`;
- В) `x->v_`;
- Г) `x.v_`;

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

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

- А) променливата `size` е недостъпна в тялото на метода;
- Б) в ред 5 не може да се променя стойността на `size_`;
- В) за метода, дефиниран на ред 4, не е определен типът на резултата;
- Г) всичко е наред;

```
1 class X {
2     const int size_;
3 public:
4     X(void){
5         size_=128;
6     }
7 };
```

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

A) конструкцията е невалидна в C++;

B) временен обект от типа `foo`;

B) променлива `bar` от тип `foo` за която се вика конструкторът по подразбиране;

Г) функция `bar` без аргументи с тип на резултата `foo`;

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

1. B
2. A
3. B
4. A

5. Г
6. B
7. Г
8. B

9. B
10. B
11. Г
12. B

13. B
14. B
15. B
16. Г