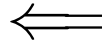


Vypni zvonenie na mobilnom telefóne!

Táto písomná skúška trvá 90 minút. Počas skúšky je povolené používať knihy, poznámky, učebné texty, referencie jazyka C/C++ v papierovej forme. Fotenie tohto dokumentu počas skúšky je zakázané. Kódy predpokladajú kompilátor s podporou C++ 14. Len jedna odpoveď A, B, C, D je správna.

Meno a priezvisko:



1. Čo sa vypíše na obrazovku, nastane chyba? (3 body) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 auto foo(int a = 1) {
4     int* b = new int;
5     *b = a;
6     return b;
7 }
8
9 int main() {
10    int* a = foo(2);
11    std::cout << *a;
12
13    delete a;
14    return 0;
15 }
```

Riešenie:

- A 1
 - B 2
 - C nastane pamäťová chyba, nedefinované správanie
 - D nedá sa skompilovať
-

2. Čo sa vypíše na obrazovku, nastane chyba? (3 body) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 auto foo(int a = 1) {
4     int* b = new int(); //int()
5     *b = a;
6     return b;
7 }
8
9 int main() {
10    int* a = foo(2);
11    std::cout << *a;
12
13    delete a;
14    return 0;
15 }
```

Riešenie:

- A 1
 - B 2
 - C nastane pamäťová chyba, nedefinované správanie
 - D nedá sa skompilovať
-

3. Čo sa vypíše na obrazovku, nastane chyba? (8 bodov) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 #include <string>
3 using std::string;
4
5 class Contract {
6 public:
7     string s = "Yello!";
8     Contract(const string& s = "Jello!"):s(s) {} //const
9 };
10
11 int main() {
12     Contract a = string("Hello!");
13     std::cout << a.s;
14
15     return 0;
16 }
```

Riešenie:

- A Yello!
- B Jello!
- C Hello!
- D nedá sa skompilovať

4. Čo sa vypíše na obrazovku, nastane chyba? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 #include <string>
3 using std::string;
4
5 class Contract {
6 public:
7     string s = "Yello!";
8     Contract(string& s = "Jello!"):s(s) {} //bez const
9 };
10
11 int main() {
12     Contract a = string("Hello!");
13     std::cout << a.s;
14
15     return 0;
16 }
```

Riešenie:

- A Yello!
 - B Jello!
 - C Hello!
 - D nedá sa skompilovať
-

5. Čo sa vypíše na obrazovku, nastane chyba? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 using std::cout;
3
4 class ContractA {
5     unsigned int ether;
6 public:
7     ContractA(unsigned int e = 0): ether(e) {}
8     auto sendEther() {
9         return ether;
10    }
11 };
12
13 class ContractB: protected ContractA { //protected
14     unsigned int wei;
15 public:
16     ContractB(unsigned int w = 1): wei(w) {}
17     auto sendWei() {
18         return wei;
19    }
20 };
21
22 int main() {
23     ContractB b;
24     cout << b.sendEther() << " " << b.sendWei();
25
26     return 0;
27 }
```

Riešenie:

- A 0 1
 - B 1 0
 - C nastane pamäťová chyba, nedefinované správanie
 - D nedá sa skompilovať
-

6. Čo sa vypíše na obrazovku? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 #include <memory>
3 #include <vector>
4 using namespace std;
5
6 class Contract {};
7
8 auto foo(shared_ptr<Contract> a) {return a;}
9
10 int main() {
11     vector<shared_ptr<Contract> > vec;
12     shared_ptr<Contract> a = make_shared<Contract>();
13
14     vec.push_back(foo(a));
15
16     cout << a.use_count();
17     return 0;
18 }
```

Riešenie:

- A 0
 - B 1
 - C 2
 - D 3
-

7. Čo sa vypíše na obrazovku, nastane chyba? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class ContractA {
4     unsigned int ether = 0;
5 public:
6     ContractA(unsigned int e = 0): ether(e) {}
7     auto sendEther() {return ether;}
8 };
9
10 class ContractB {
11     unsigned int wei = 1;
12 public:
13     ContractB(unsigned int w): wei(w) {}
14     auto sendWei() {return wei;}
15 };
16
17 int main() {
18     ContractA a;
19     ContractB b;
20     std::cout << a.sendEther() << "_" << b.sendWei();
21
22     return 0;
23 }
```

Riešenie:

- A 0 0
 - B 0 1
 - C nastane pamäťová chyba, nedefinované správanie
 - D nedá sa skompilovať
-

8. Čo sa vypíše na obrazovku? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 #include <set>
3 #include <vector>
4 using namespace std;
5
6 int main() {
7     vector<int> v = {1, 1, 1, 0, 0, 0};
8     set<int, less<int> > s;
9
10    for(auto i: v) {
11        s.insert(i);
12    }
13
14    for(auto i: s) {
15        cout << i << "_";
16    }
17
18    return 0;
19 }
```

Riešenie:

- A 0 0
- B 0 1
- C 1 0
- D 3 0

9. Čo sa vypíše na obrazovku? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm> //stable_sort
4 #include <cstdlib> //abs
5 using namespace std;
6
7 int main() {
8     vector<int> v = {-1, 0, 1};
9     auto s = v.begin();
10    auto e = v.end();
11
12    stable_sort(s, e, [](const int& a, const int& b) {
13        return abs(a) < abs(b);
14    });
15
16    for(auto a: v) {
17        cout << a << " ";
18    }
19
20    return 0;
21 }
```

Riešenie:

- A -1 0 1
 - B 0 -1 1
 - C -1 1 0
 - D 1 -1 0
-

10. Čo sa vypíše na obrazovku, nastane chyba? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 using std::cout;
3
4 class ContractA {
5 public:
6     ContractA& operator=(const ContractA& a) {
7         cout << "ContractA&_operator=(const_ContractA&);";
8         return *this;
9     }
10    ContractA(const ContractA& a) {
11        cout << "ContractA(const_ContractA&);";
12    }
13    ContractA() = default;
14 };
15
16 class ContractB: public ContractA {
17 public:
18     ContractB& operator=(const ContractB& b) {
19         cout << "ContractB&_operator=(const_ContractB&);";
20         return *this;
21     }
22    ContractB(const ContractB& b) {
23        cout << "ContractB(const_ContractB&);";
24    }
25    ContractB() = default;
26 };
27
28 void foo(ContractA& a) {}
29 void bar(ContractB& b) {}
30
31 int main() {
32     ContractA a;
33     ContractB b;
34     foo(a);
35     bar(b);
36
37     return 0;
38 }
```

Riešenie:

- A ContractA& operator=(const ContractA&)
- B ContractA(const ContractA&)
- C ContractB(const ContractB&)
- D nič sa nevypíše