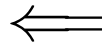


Vypni zvonenie na mobilnom telefóne!

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? (3 body)

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

```
1 #include <iostream>
2 using std::cout;
3
4 auto foo(int& n) {
5     return 1;
6 }
7 auto foo(int n) {
8     return 2;
9 }
10
11 int main() {
12     cout << foo(3);
13
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? (3 body)

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

```
1 #include <iostream>
2 using std::cout;
3
4 auto foo(int& n) {return 1;}
5 auto foo(int&& n) {return 2;}
6
7 int main() {
8     cout << foo(3);
9
10    return 0;
11 }
```

Riešenie:

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

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

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

```
1 #include <iostream>
2 using std::cout;
3
4 template <typename T>
5 T const& max(T& a, T& b) {
6     return a < b ? b : a;
7 }
8
9 int main() {
10    cout << max(39, 20);
11
12    return 0;
13 }
```

Riešenie:

- A 39
- B 20
- C nastane pamäťová chyba, nedefinované správanie
- D nedá sa skompilovať

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

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

```
1 #include <iostream>
2 using std::cout;
3
4 template <typename T>
5 T const& max(T const a, T const b) {
6     return a < b ? b : a;
7 }
8
9 int main() {
10     cout << max(39, 20);
11
12     return 0;
13 }
```

Riešenie:

- A 39
  - B 20
  - C nastane pamäťová chyba, nedefinované správanie
  - D nedá sa skompilovať
- 

5. Čo sa vypíše na obrazovku? (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: private ContractA {
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 using std::cout;
3
4 class ContractA {
5     unsigned int ether = 0;
6 public:
7     ContractA(unsigned int e): ether(e) {}
8     auto sendEther() {
9         return ether;
10    }
11 };
12
13 class ContractB: public ContractA {
14     unsigned int wei = 1;
15 public:
16     ContractB(unsigned int w): wei(w) {}
17     auto sendWei() {
18         return wei;
19    }
20 };
```

```

21
22 int main() {
23     ContractB b(0);
24     cout << b.sendEther() << " " << b.sendWei();
25
26     return 0;
27 }

```

Riešenie:

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

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

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

```

1 #include <iostream>
2 #include <map>
3 #include <vector>
4 using std::cout;
5
6 class Contract {
7 public:
8     unsigned int id = 1;
9 };
10
11 int main() {
12     std::map<unsigned int, std::vector<Contract*>>* m =
13         new std::map<unsigned int, std::vector<Contract*>>;
14     Contract* b = new Contract;
15     std::vector<Contract*>* v = new std::vector<Contract*>;
16
17     (*m)[b->id] = v;
18     v->push_back(b);
19     cout << ((*m).at(b->id))->back(); //back() vráti posledný prvok
20
21     delete v;
22     delete b;
23     delete m;
24     return 0;
25 }

```

Riešenie:

- A 1
  - B adresa
  - 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 <map>
3 #include <string>
4 using std::cout;
5
6 int main() {
7     std::map<unsigned int, std::string> m;
8
9     try {
10         m.at(0) = "tokyo";
11     } catch(...) {
12         m.at(0) = "kyoto";
13     }
14
15     cout << m.at(0);
16
17     return 0;
18 }

```

Riešenie:

- A tokyo
- B kyoto
- C nezachytená výnimka std::bad\_alloc
- D nezachytená výnimka std::out\_of\_range

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

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

```
1 #include <iostream>
2 using std::cout;
3
4 class Contract {
5     unsigned int id = 1;
6 public:
7     Contract(unsigned int id = 0): id(id) {
8         id = 2;
9     }
10 };
11
12 class Token {
13     friend class Contract;
14 public:
15     unsigned int getContractId(const Contract& b) {
16         return b.id;
17     }
18 };
19
20 int main() {
21     Contract b;
22     Token t;
23     cout << t.getContractId(b);
24
25     return 0;
26 }
```

Riešenie:

- A 0
- B 1
- C 2
- D nedá sa skompilovať

10. Čo sa vypíše na obrazovku? (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