# Písomná skúška z PT: 1. februára 2021

**Táto písomná skúška trvá 90 minút. Počas skúšky je povolené používať knihy, poznámky, učebné texty, referencie jazyka C/C++, kompilátor jazyka C/C++. Kódy predpokladajú kompilátor s podporou C++ 17. Túto písomnú skúšku je potrebné vypracovať samostatne.**

**1. Prečo kompilátor nevygeneruje kopírovací konštruktor? (8 b.)**

```
#include <iostream>

class Token {
public:
  Token() = default;
  Token(Token&& t) { std::cout << "Token(Token&&)"; }
};

int main() {
  Token t0;
  Token t1 = t0;

  return 0;
}
```

---

**2. Prečo sa zavolá kopírovací konštruktor, napriek tomu, že voláme std::move(t0)? Prečo kompilátor nevygeneruje presúvací konštruktor? (8 b.)**

```
#include <iostream>

class Token {
public:
  Token() = default;
  Token(const Token& t) { std::cout << "Token(const␣Token&)"; }
};

int main() {
  Token t0;
  Token t1 = std::move(t0);

  return 0;
}
```

---

**3. Prečo kód neskompiluje? (4 b.)**

```
#include <string>

int main() {
  char str0[20] = "hello␣world!";
  str0[0] = 'H';

  std::string str1 = "hello␣world!";
  char* str2 = str1.c_str();
  str2[0] = 'H';

  return 0;
}
```

---

**4. Prečo kód neskompiluje? (8 b.)**

```
class TokenB{
public:
  int a{2};
  TokenB operator*(const TokenB& t0) {
    TokenB t;
    t.a = a * t0.a;

    return t;
  }
};

class TokenD : public TokenB {};

int main() {
  TokenB t0, t1;
  TokenB t2 = t0 * t1;

  TokenD t3, t4;
  TokenD t5 = t3 * t4;

  return 0;
}
```

**5. Prečo kód neskompiluje? (8 b.)**

```
#include <iostream>

class Token{
public:
  int a{0};
  Token(int a) { a = -a; }
};

int main() {
  Token t0(int(1));
  std::cout << t0.a;

  Token t1(int());
  std::cout << t1.a;

  return 0;
}
```

**6. Prečo kód neskompiluje? (8 b.)**

```
class Token {
public:
  int a{0};
  bool operator<(const Token& t0) {
    return this->a < t0.a;
  }
};

template <typename T>
T max(const T& a, const T& b) {
  return a<b ? b : a;
}

int main() {
  Token t0, t1;
  Token t2 = max(t0, t1);

  return 0;
}
```

**7. Prečo potrebujeme lambda funkciu cmp? Akú dátovú štruktúru využíva std::set? (8 b.)**

```
#include <set>

class Token {
public:
  int a{0};
};

int main() {
  auto cmp = [](const Token& t0, const Token& t1) {
               return t0.a < t1.a;
             };

  std::set<Token, decltype(cmp) > s(cmp);
  Token t0;

  s.insert(t0);

  return 0;
}
```

**8. Prečo potrebujeme lambda funkcie hash a equal? (8 b.)**

```
#include <unordered_set>

class Token {
public:
  int a{0};
};

int main() {
  auto hash = [](const Token& t0) {
                return t0.a % 11;
              };

  auto equal = [](const Token& t0, const Token& t1) {
                return t0.a == t1.a;
              };

  std::unordered_set<Token, decltype(hash), decltype(equal) > s(11, hash, equal);
  Token t0;

  s.insert(t0);

  return 0;
}
```

---