

1.

1) `struct A {`
`A() { std::cout << "Constr "; }`
`A(const A & lvr) {} // Если перегружают также A(A&&){} – норм.`
`A& operator=(const A && a) { return *this; }`
`A& operator=(const A & a) { return *this; }`
`};`
`int main() {`
`try {`
`A a, b(a); // Err! Запрещено копирование`
`a = b; // Err! Запрещена операция присваивания`
`throw a; // Err! Запрещены конструкторы переноса и копирования`
`}`
`catch (...) { std::cout << "Catch(...) \n"; } }`
2) Или удалить операцию переноса

Критерии: За каждую найденную, необъясненную, лишнюю ошибку - -3
За отсутствие/неверное исправление каждой ошибки - -2

2.

`int main() {`
`using namespace NS1;`
`using ::a (или NS1::a/NS2::a); или описать локальную переменную, а`
`f(a, "abc"); // NS2::f()`
`f(true, 1.5f); // NS1::f()`
`}`

Критерии: За найденную, лишнюю ошибку - -5, неверное исправление - -3
За каждую неверную печать - -3

3.

<code>class B {</code> <code>public:</code> <code> void a () const { std::cout << "1 "; }</code> <code> virtual void b () const { std::cout << "2 "; }</code> <code> virtual ~B () { std::cout << "3 "; }</code> <code>};</code>	<code>void handle(const B& rb) {</code> <code> rb.a();</code> <code> rb.b();</code> <code> if (typeid(rb) == typeid(D)) {</code> <code> dynamic_cast<const D&>(rb).c();</code> <code> }</code> <code>}</code>
<code>class D : public B {</code> <code>public:</code> <code> void b () const override {std::cout << "4 "; }</code> <code> void c () const { std::cout << "5 "; }</code> <code> ~D () override { std::cout << "6 "; }</code> <code>};</code>	<code>int main() {</code> <code> handle(B()); // 1 2 3</code> <code> std::cout << endl;</code> <code> handle(D()); // 1 4 5 6 3</code> <code> std::cout << endl;</code> <code>}</code>

Критерии: За каждый неуместный (не приводящий к требуемой печати), лишний метод - -4.
За пропуск **const** в методах - -4 (за все),
Если прототипы виртуальных методов отличаются квалификатором **const** + еще -4.

4. `template <class T>`

`typename T::iterator f (T & cont) {//везде может быть const iterator`
`typename T::iterator p = cont.begin(), pmin = p;`
`while (++p != cont.end()) {`
`if (*p < *pmin) pmin = p;`
`}`
`return pmin; }`

Критерии: За передачу контейнера по значению- -5 .
За отсутствие **typename**- -5 (за все).
За другие ошибки - -3 за каждую.

5. Да, если в нем описана своя виртуальная функция

Критерии: нет/неверное обоснование - -10.

6.

$S \rightarrow t \wedge S \mid x \wedge A \mid f \langle \text{cout} \ll 'f'; \rangle$
 $A \rightarrow x \wedge A \mid f \langle \text{cout} \ll 'f'; \rangle \mid t \langle \text{cout} \ll 'x'; \rangle$

Критерии: за верный перевод ПОЛИЗ, но не оптимальный по длине: **-5** ;
за каждое неверное/отсутствующее действие по переводу: **-5**.

7. (а) $\text{first}(Ca) \cap \text{first}(C) = \{c\} \neq \emptyset$
или $\text{first}(Bb) \cap \text{first}(b) = \{b\} \neq \emptyset$

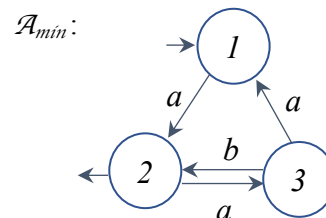
(б) $S \rightarrow ABC$ (возможны другие варианты)
 $A \rightarrow CX$
 $X \rightarrow a \mid \varepsilon$
 $B \rightarrow bY$
 $Y \rightarrow bY \mid \varepsilon$
 $C \rightarrow c$

Критерии: (а) за отсутствующее/неверное обоснование **-5**.
(б) за отсутствующую/неэквивалентную или не удовлетворяющую критерию грамматику **-5**.

8. $S \rightarrow AA$ (возможны другие варианты)
 $AA \rightarrow aa$

Критерии: за неверный язык: **-5**.
за несоответствие грамматики по каждому пункту: **-3**

9. Для получения минимального ДКА можно воспользоваться алгоритмом Бржозовского, или, поскольку исходный автомат детерминированный – алгоритмом Хопкрофта. С точностью до названий состояний минимальный ДКА единственный.



Критерии: за отсутствие обоснования при верном ответе, или наличие вычислительных ошибок при исполнении шагов алгоритмов, приведших к искаженному результату: **-5**.

10. *Программный продукт* – программа, которая работает без авторского надзора в рамках некоторого набора операционных сред. Может исполняться, тестироваться и модифицироваться без участия автора.

Критерии: **-10** за ошибочное/отсутствующее определение. (Достаточно передать смысл понятия своими словами. Ключевой момент – для использования ПП автор не нужен, продукт отчужден от автора).