

1. Varijable u namespaceovima se tretiraju kao globalne I one imaju vrijednost 0 ako nisu inicijalizirane.
2. Sintaksa importovanja svih simbola iz namespacea je **using namespace ime_namespacea;** Kada importujemo sve simbole, tada ce oni biti vidljivi main funkciji, a ukoliko zelimo da importujemo pojedinačno simbole kositimo sljedecu sintaksu: **using ime_namespacea::ime_varijable;** I tada je samo taj symbol koji importujemo vidljiv.
3. Ugniježdeni namespace je zapravo namespace u namespaceu – Ugniježdeni namespace je prikazana na sljedecoj slici:

```
namespace A {  
    int a=5;  
    namespace B  
    {  
        int b=10;  
    }  
}
```

Ukoliko zelimo da importujemo sadržaj namespacea A, koristit cemo sljedecu sintaksu:
using namespace A; ovako cemo dobiti varijablu a=5.

Medjutim ukoliko sada zelimo da importujemo sadržaj namespacea B, to radimo koristeći sintaksu: using namespace A::B;

Takodje ako hocemo da importujemo pojedinačno symbol iz namespacea B, to radimo na sljedeci način: using A::B::b;

4. Razlika pristupanja pomoću operatora [] i metode at() je u tome što kada koristimo operator [] on provjerava granice containera I ako predjemo granicu program ce imati nedefinisano ponašanje, dok at() to ne radi I ako koristeći ovaj metod prekoracimo granicu program nam baca iznimku I upozorenje da smo granicu prekoracili.
5. Clear metoda nad vektorom – brise sve elemente vektora, odnosno prazni ga.
6. Getline je funkcija koja omogucava prihvatanje I citanje pojedinačnih I višestrukih nizova iz ulaznog toka.
7. Buffer je područje memorije koje se koristi za privremeno memorisanje podataka dok se premjestaju s jednog mjesta na drugo.
8. Escape karakteri se koriste da reprezentuju posebne znakove unutar niza I karaktera. Npr. “\n” – prelazak u novi red. Takodje prilikom korištenja istream funkcije cin, u buffer se smjesta “\n”, npr ukoliko hocemo da unesemo 3, u buffer ce se spremi “3\n”.

9. Nevalidan unos za `std::cin` predstavlja unos koji nije istog tipa kao deklarirana varijabla za unos. Npr. ako imamo sljedeće:

```
int x{};
std::cin>>x;
```

Ukoliko unesemo npr. slovo b, unos će biti nevalidan zato što slovo b se ne može ekstrahovati u `int` varijablu.

10. EOF je makro definicija tipa `int` koja se proširuje u negativni integralni konstantni izraz (općenito, -1). Koristi se kao vrijednost koju vraća nekoliko funkcija u zaglavlju `<cstdio>` za označavanje da je dosegnut kraj fajla ili za signaliziranje nekih drugih stanja kvara. Također se koristi kao vrijednost za predstavljanje nevažećeg znaka. U C++, ova makronaredba odgovara vrijednosti `char_traits<char>::eof()`.

11. U C++, scope resolution operator je `::`. Koristi se u sljedeće svrhe.

- a) Za pristupanje za pristupanje globalnoj varijabli ako imamo i lokalnu varijablu s istim imenom.
- b) Za definisanje funkcije van klase
- c) Za pristupanje statičkim varijablama koje se nalaze u klasama
- d) U slučaju višestrukog nasljeđivanja: Ako isto ime varijable postoji u dvjema klasama predaka, možemo koristiti operator scope resolution za razlikovanje.
- e) Za namespace: Ako a klasa ima isto ime unutar dva namespacea možemo koristiti ime namespacea sa scope resolution operatorom da se uputimo na tu klasu bez problema.
Npr: `namespace A::b;`

d)

Upućivanje na klasu unutar druge klase: Ako klasa postoji unutar druge klase, možemo upotrijebiti ugniježđenu klasu za upućivanje na ugniježđenu klasu pomoću operatora rezolucije opsega (scope resolution operator).