

Inline funkcije

- ❖ Funkcija se može deklarirati kao *inline* navođenjem ovog specifikatora u deklaraciji. Funkcije članice klase *X*, kao i prijateljske funkcije klasi *X* koje su definisane u definiciji date klase *X* (u definiciji klase im je navedeno i telo) su implicitno *inline*, čak i ako se to ne naglasi:

```
inline void strcpy(char* to, const char* from) { while (*to++ = *from++); }
```

```
class Person {  
public:  
    string getName () const { return name; }  
    Person& setName (const string& newName) { name = newName; return *this; }  
    ...  
};
```

Ove funkcije su implicitno *inline*

ili:

```
class Person {  
public:  
    inline string getName () const;  
    inline Person& setName (const string& newName);  
    ...  
};  
  
string Person::getName () const { return name; }  
Person& Person::setName (const string& newName) { name = newName; return *this; }
```

Inline funkcije

- ❖ Da bi ugradio kod *inline* funkcije na mestu poziva, prevodilac mora da ima kompletnu definiciju (tela) te funkcije. Zato su za ove funkcije dozvoljene *višestruke definicije* u istom programu, ali samo po jedna u svakoj jedinici prevođenja; po pravilu, definicije *inline* funkcija navode se u zaglavljima
- ❖ Čak i ako se nalaze u različitim jedinicama prevođenja, višestruke definicije *inline* funkcija odnose se na istu funkciju i semantika te funkcije se ne menja: ona će imati istu adresu, lokalni statički objekti su jedinstveni za sve definicije (sve iste definicije takvog objekata odnose se na jedan jedinstveni entitet) i slično
- ❖ Sve definicije iste *inline* funkcije u različitim jedinicama prevođenja moraju biti identične; ako nisu, ponašanje programa je nedefinisano
- ❖ Sa jedne strane, prevodilac ne mora ispoštovati zahtev za ugrađivanje *inline* funkcije u kod pozivaoca, već generisati kod za uobičajen poziv. Sa druge strane, ne brani se prevodiocu da kod i neke druge funkcije koja nije *inline* ugradi na mesto poziva kao optimizaciju. Zbog toga se osnovni smisao *inline* funkcija zapravo gubi, pa *inline* formalno ne znači ništa više nego to da su dozvoljene višestruke definicije iste funkcije u različitim jedinicama prevođenja u istom programu
- ❖ Zbog toga je u jezik uveden i koncept *inline* statičkih varijabli (objekata ili referenci): dozvoljene su višestruke definicije *inline* statičke varijable u različitim jedinicama prevođenja istog programa (ali samo po jedna definicija u jednoj jedinici), pod uslovom da su te definicije identične; sve one se odnose na isti entitet
- ❖ Ovo omogućava kompletne definicije biblioteka u fajlovima zaglavljima, bez potrebe za definisanje objekata u *.cpp* fajlovima

inline istream cout;