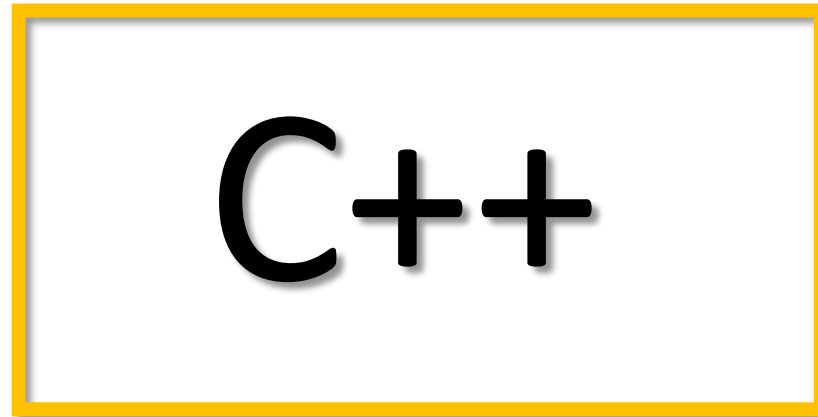


# Uvod u objektno – orjentisano programiranje



Prof. Miloš Mojović

# Program „Hello World“

- Mora se od nečega početi. Najpre instaliramo program „codeblocks-20.03mingw-setup.exe“
- <https://www.youtube.com/watch?v=EmPdhDE-wrl&t=1s>
- Namestimno da nam podešavanja za kompajler budu „Default“:  
Settings>Compiler>GlobalCompilerSettings>Toolchain executables>AutoDetect>OK.
- File>New>Project>Console application>C++ (Ostalo Default)>Sources>main.cpp

```
#include <iostream> /*Preprocesorska komanda*/  
  
int main()  
{  
    std::cout << "Cao studenti!" << std::endl;  
    return 0;  
}
```

```
#include <iostream> /*Preprocesorska komanda*/  
  
using namespace std;  
  
int main()  
{  
    cout << "Cao studenti!";  
    return 0;  
}
```

Cao studenti!

# Program „Hello World“

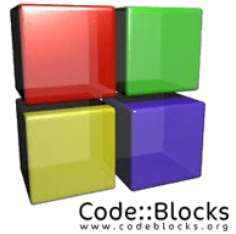


- Mora se od nečega početi. Najpre instališimo program VS
- [https://www.youtube.com/watch?v=Q\\_7dFpjimbg&t=549s](https://www.youtube.com/watch?v=Q_7dFpjimbg&t=549s)
- Create New Project > (C++ ; Windows ; Console) > Empty Project C++ > Create
- Source Files > Add > New Item > C++ File
- Cekirati: Tools>Options>Debugging>General>Automatically close the console when debugging stops da vas ne smara sa obavestjenjem

```
#include <iostream> /*Pretprocesorska komanda za broj*/  
  
int main()  
{  
    std::cout << "Cao studenti!" << std::endl;  
    std::cin.get(); /*Ceka novi unos pa pazira program*/  
}
```

```
#include <iostream> /*Pretprocesorska komanda za broj*/  
  
using namespace std;  
  
int main()  
{  
    cout << "Cao studenti!";  
    cin.get(); /*Ceka novi unos pa pazira program*/  
}
```

```
Cao studenti!
```



# Program „Prikaz unetog broja“

Napravimo program koji nam najpre traži unos a zatim prikazuje uneti broj:

```
#include <iostream>
using namespace std;

int main()
{
    int number;

    cout << "Unesi neki broj: ";
    cin >> number;

    cout << "Uneli ste broj " << number;
    return 0;
}
```

```
Unesi neki broj: 3
Uneli ste broj 3
```

# Program „Prikaz unetog broja“



Visual Studio

Napravimo program koji nam najpre traži unos a zatim prikazuje uneti broj:

```
#include <iostream> /*pretprocesorska komanda*/
using namespace std; /*deklarisanje*/

int main() /*funkcija main*/
{
    int number;
    cout << "Unesi neki broj: "; /*console out*/
    cin >> number; /*console in*/
    cout << "Uneli ste broj " << number;
    cout << "\n"; /*novi red*/
    system("pause"); /*pauza*/
    return 0; /*kraj programa*/
}
```

```
Unesi neki broj: 3
Uneli ste broj 3
```

# Program „Suma dva broja“



Napravimo program koji nam najpre traži unos dva broja, a zatim prikazuje njihovu sumu:

```
#include <iostream>
using namespace std;

int main()
{
    int prviBroj, drugiBroj, sumaDvaBroja;

    cout << "Unesi dva broja: ";
    cin >> prviBroj >> drugiBroj;

    // pravi novu promenljivu koja je sumadva broja
    sumaDvaBroja = prviBroj + drugiBroj;

    // prikazuje sumu
    cout << prviBroj << " + " << drugiBroj << " = " << sumaDvaBroja;

    return 0;
}
```

```
Unesi dva broja: 2
4
2 + 4 = 6
```

# Program „Suma dva broja“ – malo lepše



Code::Blocks  
www.codeblocks.org

Napravimo program koji nam najpre traži unos dva broja, a zatim prikazuje njihovu sumu:

```
#include <iostream>
using namespace std;

int main()
{
    int prviBroj, drugiBroj, sumaDvaBroja;
    cout << "Unesi dva broja";
    cout << "\n"; /*Novi red*/
    cout << "Prvi broj: ";cin >> prviBroj;
    cout << "Drugi broj: ";cin >> drugiBroj;
    // pravi novu promenljivu koja je sumadva broja
    sumaDvaBroja = prviBroj + drugiBroj;
    // prikazuje sumu
    cout << prviBroj << " + " << drugiBroj << " = " << sumaDvaBroja;
    return 0;
}
```

```
Unesi dva broja
Prvi broj: 2
Drugi broj: 4
2 + 4 = 6
```

# Program „Suma dva broja“



Napravimo program koji nam najpre traži unos dva broja, a zatim prikazuje njihovu sumu:

```
#include <iostream>
using namespace std;

int main()
{
    int prviBroj, drugiBroj, sumaDvaBroja;

    cout << "Unesi dva broja: ";
    cin >> prviBroj >> drugiBroj;

    // pravi novu promenljivu koja je suma dva broja
    sumaDvaBroja = prviBroj + drugiBroj;

    // prikazuje sumu
    cout << prviBroj << " + " << drugiBroj << " = " << sumaDvaBroja;

    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/

    return 0;
}
```

```
Unesi dva broja: 2
4
2 + 4 = 6
```



# Program „Unos broja i imena“



Napravimo program koji nam traži unos broja i imena, i na kraju prikazuje njihov ispis:

```
#include <iostream> /*Pretprocesorska komanda za broj*/
#include <string> /*Pretprocesorska komanda za rec*/
using namespace std; /*Kaze da ce se na dalje koristiti standardni imenski prostor*/

int main() /*Poziva funkciju koja vraća ceo broj*/
{
    int unosBroja; /*Pravi novu promenljivu - broj, pod nazivom unosBroja*/
    cout << "Unesi neki broj: "; /*Ispis na ekranu*/
    cin >> unosBroja; /*Trazi unos promenljive - broja, pod nazivom unosBroja*/

    string unosImena; /*Pravi novu promenljivu - string, pod nazivom unosImena*/
    cout << "Unesi svoje ime: "; /*Ispis na ekranu*/
    cin >> unosImena; /*Trazi unos promenljive - reci, pod nazivom unosImena*/

    cout << unosImena << " je uneo broj " << unosBroja << endl; /*Ispis na ekranu*/
    return 0; /*Vraća broj funkcije main - 0 za uspešno urađen program*/
}
```

- Probajte da unesete više reči. Da li program funkcioniše?

cout – console out  
cin – console in

```
Unesi neki broj: 3
Unesi svoje ime: Pera
Pera je uneo broj 3
```

# Program „Unos broja i imena“



Napravimo program koji nam traži unos broja i imena, i na kraju prikazuje njihov ispis:

```
#include <iostream> /*Pretprocesorska komanda za broj*/
#include <string> /*Pretprocesorska komanda za rec*/
using namespace std; /*Kaze da ce se na dalje koristiti standardni imenski
prostor*/

int main() /*Poziva funkciju koja vraća ceo broj*/
{
    int unosBroja; /*Pravi novu promenljivu - broj, pod nazivom unosBroja*/
    cout << "Unesi neki broj: "; /*Ispis na ekranu*/
    cin >> unosBroja; /*Trazi unos promenljive - broja, pod nazivom unosBroja*/

    string unosImena; /*Pravi novu promenljivu - string, pod nazivom
unosImena*/
    cout << "Unesi svoje ime: "; /*Ispis na ekranu*/
    cin >> unosImena; /*Trazi unos promenljive - reci, pod nazivom unosImena*/

    cout << unosImena << " je uneo broj " << unosBroja << endl; /*Ispis na
ekranu*/
    system("pause"); /*Pauza*/
    return 0; /*Vraca broj funkcije main - 0 za uspesno uradjen program*/
}
```

- Probajte da unesete više reči. Da li program funkcioniše?

cout – console out  
cin – console in

```
Unesi neki broj: 3
Unesi svoje ime: Pera
Pera je uneo broj 3
```

# Program „Samoglasnik ili suglasnik“



Napravimo program koji nam traži unos nekog slova te nam kaže da li je to samoglasnik ili suglasnik:

```
#include <iostream>
using namespace std;

int main()
{
    char c;
    int maloSlovo, velikoSlovo;

    cout << "Unesi neko slovo: ";
    cin >> c;

    // postavlja vrednost promenljive maloSlovo da bude 1 (true) ako je c samoglasnik pisan malim slovom
    // znak || znaci "ili"
    maloSlovo = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

    // postavlja vrednost promenljive velikoSlovo da bude 1 (true) ako je c samoglasnik pisan velikim slovom
    velikoSlovo = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');

    // proverava da li je naka od promenljivih maloSlovo ili velikoSkovo 1 (true)
    if (maloSlovo || velikoSlovo)
        cout << c << " je samoglasnik.";
    else
        cout << c << " je suglasnik.";

    return 0;
}
```

Pravi jednu karakternu promenjivu: c

**Karakterna promenljiva** moze biti samo jedan karakter koji je ili: broj, slovo, simbol ili prazan proctor. Zauzima do 8 bitova (1 bajt).

Pravi dve brojevne promenljive: maloSlovo i velikoSlovo

Unesi neko slovo: s  
s je suglasnik.

Unesi neko slovo: A  
A je samoglasnik.

# Program „Samoglasnik ili suglasnik“ (da bude jasnije)



Napravimo program koji nam traži unos nekog slova te nam kaže da li je to samoglasnik ili suglasnik:

```
#include <iostream>
using namespace std;
int main()
{
    char c;
    int maloSlovo, velikoSlovo;
    cout << "Unesi neko slovo: ";
    cin >> c;
    // postavlja vrednost promenljive maloSlovo da bude 1 (true) ako je c samoglasnik pisan malim slovom
    // znak || znaci "ili"
    maloSlovo = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
    // postavlja vrednost promenljive velikoSlovo da bude 1 (true) ako je c samoglasnik pisan velikim slovom
    velikoSlovo = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
    // proverava da li je naka od promenljivih maloSlovo ili velikoSkovo 1 (true)
    if (maloSlovo || velikoSlovo == 1)
        cout << c << " je samoglasnik.";
    else
        cout << c << " je suglasnik.";
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

Pravi jednu karakternu promenjivu: c

**Karakterna promenljiva** moze biti samo jedan karakter koji je ili: broj, slovo, simbol ili prazan proctor. Zauzima do 8 bitova (1 bajt).

Pravi dve brojevne promenljive: maloSlovo i velikoSlovo

Unesi neko slovo: s  
s je suglasnik.

Unesi neko slovo: A  
A je samoglasnik.

# Program „Samoglasnik ili suglasnik“



Napravimo program koji nam traži unos nekog slova te nam kaže da li je to samoglasnik ili suglasnik:

```
#include <iostream>
using namespace std;
int main()
{
    char c;
    int maloSlovo, velikoSlovo;
    cout << "Unesi neko slovo: ";
    cin >> c;
    // postavlja vrednost promenljive maloSlovo da bude 1 (true) ako je c samoglasnik pisan malim slovom
    // znak || znaci "ili"
    maloSlovo = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
    // postavlja vrednost promenljive velikoSlovo da bude 1 (true) ako je c samoglasnik pisan velikim slovom
    velikoSlovo = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
    // proverava da li je naka od promenljivih maloSlovo ili velikoSkovo 1 (true)
    if (maloSlovo || velikoSlovo)
        cout << c << " je samoglasnik.";
    else
        cout << c << " je suglasnik.";
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

Pravi jednu karakternu promenjivu: c  
Pravi dve brojevne promenljive:  
maloSlovo i velikoSlovo

Unesi neko slovo: s  
s je suglasnik.

Unesi neko slovo: A  
A je samoglasnik.

# Program „Par nepar“



Napravimo program koji nam traži unos nekog sbroja te nam program kaže da li je to paran ili neparan broj:

```
#include <iostream>
using namespace std;

int main()

// Program koji kaze da li je uneti broj paran ili neparan
{
    int n;

    cout << "Unesi neki broj: ";
    cin >> n;

    if ( n % 2 == 0)

// Komanda % (modulus operator) prikazuje ostatak pri deljnju sa nekim brojem

        cout << n << " je paran broj.";
    else
        cout << n << " je neparan broj.";

    return 0;
}
```

Unesi neki broj: 4  
4 je paran broj.

# Program „Par nepar“



Napravimo program koji nam traži unos nekog sbroja te nam program kaže da li je to paran ili neparan broj:

```
#include <iostream>
using namespace std;
int main()
// Program koji kaze da li je uneti broj paran ili neparan
{
    int n;
    cout << "Unesi neki broj: ";
    cin >> n;
    if (n % 2 == 0)
        // Komanda % (modulus operator) prikazuje ostatak pri
deljnju sa nekim brojem
        cout << n << " je paran broj.";
    else
        cout << n << " je napanan broj";
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

```
Unesi neki broj: 4
4 je paran broj.
```

# Množenje 2 broja



Napravimo program koji nam traži unos nekog 2 broja te nam da ispis o njihovom množenju:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Unesi prvi broj" << endl; /*Ispisuje da trazi unos prvog broja i ide u novi red (endl)*/
    int prviBroj = 0; /*Pravi promenjivu koja je broj i postavlja joj inicijalnu vrednost da bude = 0*/
    cin >> prviBroj; /*Trazi unos promenjive*/

    cout << "Unesi drugi broj" << endl;
    int drugiBroj = 0;
    cin >> drugiBroj;

    int proizvodBrojeva = prviBroj * drugiBroj; /*Pravi promenjivu koja je proizvod dve unente
    promenljive*/

    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva << endl; /*Ispisuje proizvod*/

    return 0;
}
```

```
Unesi prvi broj
2
Unesi drugi broj
3
2*3=6
```



# Množenje 2 broja



Napravimo program koji nam traži unos nekog 2 broja te nam da ispis o njihovom množenju:

```
#include <iostream>
using namespace std;
int main()
{
    int prviBroj, drugiBroj;
    cout << "Unesi prvi broj" << endl; /*Ispisuje da trazi unos prvog broja i ide u novi red (endl)*/
    cin >> prviBroj; /*Trazi unos promenjive*/
    cout << "Unesi drugi broj" << endl;
    cin >> drugiBroj;
    int proizvodBrojeva = prviBroj * drugiBroj; /*Pravi promenjivu koja je proizvod dve unente
promenljive*/
    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva <<endl; /*Ispisuje proizvod*/
    return 0;
}
```

```
Unesi prvi broj
2
Unesi drugi broj
3
2*3=6
```

# Množenje 2 broja



Napravimo program koji nam traži unos nekog 2 broja te nam da ispis o njihovom množenju:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Unesi prvi broj" << endl; /*Ispisuje da trazi unos prvog
broja i ide u novi red (endl)*/
    int prviBroj = 0; /*Pravi promenjivu koja je broj i postavlja joj
inicijalnu vrednost da bude = 0*/
    cin >> prviBroj; /*Trazi unos promenjive*/
    cout << "Unesi drugi broj" << endl;
    int drugiBroj = 0;
    cin >> drugiBroj;
    int proizvodBrojeva = prviBroj * drugiBroj; /*Pravi promenjivu koja
je proizvod dve unente promenljive*/
    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva <<
endl; /*Ispisuje proizvod*/
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

```
Unesi prvi broj
2
Unesi drugi broj
3
2*3=6
```

# Pravljenje fje za množenje 2 broja



Napravimo program koji u sebi ima posebnu fju čija je svrha množenje 2 broja:

```
#include <iostream>
using namespace std;
void FjaProizvoda() /*Pravi novu funkciju koja se zove fjaProizvoda*/
{
    cout << "Unesi prvi broj" << endl; /*Ispisuje da trazi unos prvog broja i ide u novi red (endl)*/
    int prviBroj = 0; /*Pravi promenjivu koja je broj i postavlja joj inicijalnu vrednost da bude = 0*/
    cin >> prviBroj; /*Trazi unos promenjive*/

    cout << "Unesi drugi broj" << endl;
    int drugiBroj = 0;
    cin >> drugiBroj;

    int proizvodBrojeva = prviBroj * drugiBroj; /*Pravi promenjivu koja je proizvod dve unente promenljive*/

    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva <<endl; /*Ispisuje proizvod*/
}

int main()
{
    FjaProizvoda(); /*Koristi tj. pokreće gore napravljenu fjuProizvoda*/

    /* U okviru fje main () sada ne mozemo koristiti promenljive prviBroj, drugiBroj i proizvodBrojeva jer
    nisu globalne za fju main i fjaProizvoda*/

    return 0;
}
```

```
Unesi prvi broj
2
Unesi drugi broj
3
2*3=6
```

# Pravljenje fje za množenje 2 broja



Napravimo program koji u sebi ima posebnu fju čija je svrha množenje 2 broja:

```
#include <iostream>
using namespace std;
void FjaProizvoda() /*Pravi novu funkciju koja se zove fjaProizvoda*/
{
    cout << "Unesi prvi broj" << endl; /*Ispisuje da trazi unos prvog broja i ide u novi red (endl)*/
    int prviBroj = 0; /*Pravi promenjivu koja je broj i postavlja joj inicijalnu vrednost da bude = 0*/
    cin >> prviBroj; /*Trazi unos promenjive*/
    cout << "Unesi drugi broj" << endl;
    int drugiBroj = 0;
    cin >> drugiBroj;
    int proizvodBrojeva = prviBroj * drugiBroj; /*Pravi promenjivu koja je proizvod dve unente
promenljive*/
    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva << endl; /*Ispisuje proizvod*/
}
int main()
{
    FjaProizvoda(); /*Pokreće gore napravljenu funkciju FjaProizvoda*/
    /* U okviru fje main () sada ne mozemo koristiti promenljive prviBroj, drugiBroj i proizvodBrojeva
jer
    nisu globalne za fju main i fjaProizvoda*/
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

```
Unesi prvi broj
2
Unesi drugi broj
3
2*3=6
```

# Pravljenje globalnih promenljivih kod fje za množenje 2 broja



```
#include <iostream>
using namespace std;

/*Pravimo sada 3 globalne promenljive*/
int prviBroj = 0;
int drugiBroj = 0;
int proizvodBrojeva = 0;

void FjaProizvoda() /*Pravi novu funkciju koja se zove FjaProizvoda*/
{
    cout << "Unesi prvi broj" << endl; /*Ispisuje da trazi unos prvog broja i ide u novi red (endl)*/
    cin >> prviBroj; /*Trazi unos promenljive*/

    cout << "Unesi drugi broj" << endl;
    cin >> drugiBroj;

    proizvodBrojeva = prviBroj * drugiBroj;

    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva << endl; /*Ispisuje proizvod*/
}

int main()
{
    FjaProizvoda(); /*Koristi tj. pokreće gore napravljenu FjuProizvoda*/

    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva << endl;

    /*Opet ispisuje proizvod
    brojeva da bi se pokazalo da su sada ove promenljive globalne i da važe za obe fje. Ova linija
    kada bi se ubacila u prethodni program na ovo mesto ne bi radila*/

    return 0;
}
```

- Omogućimo sada, da prethodni program poseduje globalne promenljive koje će funkcionisati sa obe funkcije (main i FjaProizvoda).
- Zapazimo kako po nekom dogovoru imenujemo funkcije (velika slova na početku reči) a kako promenljive (prvo malo a posle veliko slovo).
- Zapazimo dvostuki ispis.

```
Unesi prvi broj
2
Unesi drugi broj
3
2*3=6
2*3=6
```

# Pravljenje globalnih promenljivih kod fje za množenje 2 broja

```
#include <iostream>
using namespace std;
/*Pravimo sada 3 globalne promenljive*/
int prviBroj = 0;
int drugiBroj = 0;
int proizvodBrojeva = 0;
void FjaProizvoda() /*Pravi novu funkciju koja se zove FjaProizvoda*/
{
    cout << "Unesi prvi broj" << endl; /*Ispisuje da trazi unos prvog broja i
ide u novi red (endl)*/
    cin >> prviBroj; /*Trazi unos promenljive*/
    cout << "Unesi drugi broj" << endl;
    cin >> drugiBroj;
    proizvodBrojeva = prviBroj * drugiBroj;
    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva << endl;
/*Ispisuje proizvod*/
}
int main()
{
    FjaProizvoda(); /*Koristi tj. pokreće gore napravljenu FjuProizvoda*/
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    cout << prviBroj << "*" << drugiBroj << "=" << proizvodBrojeva << endl;
/*Opet ispisuje proizvod
brojeva da bi se pokazalo da su sada ove promenljive globalne i da važe za
obe fje. Ova linija
kada bi se ubacila u prethodni program na ovo mesto ne bi radila*/
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

- Omogućimo sada, da prethodni program poseduje globalne promenljive koje će funkcionisati sa obe funkcije (main i FjaProizvoda).
- Zapazimo kako po nekom dogovoru imenujemo funkcije (velika slova na početku reči) a kako promenljive (prvo malo a posle veliko slovo).
- Zapazimo dvostuki ispis.

Unesi prvi broj

3

Unesi drugi broj

4

3\*4=12

Press any key to continue . . .

3\*4=12

Press any key to continue . . .

# Tipovi promenljivih

bool	true ili false
char	vrednosti od 256 karaktera
unsigned short int	0 do 65.535
short int	-32.768 do 32.767
unsigned long int	0 do 4.294.967.295
long int	-2.147.483.648 do 2.147.483.647
unsigned long long	0 do 18.446.744.073.709.551.615
long long	-9.223.372.036.854.775.808 do 9.223.372.036.854.775.807
int (16 bit)	-32.768 do 32.767
int (32 bit)	-2.147.483.648 do 2.147.483.647
unsigned int (16 bit)	0 do 65.535
unsigned int (32 bit)	0 do 4.294.967.295
float	1.2e-38 do 3.4e38
double	2.2e-308 do 1.8e308

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    //float pi=3.14;
```

```
    //int pi=3.14;
```

```
    cout << pi;
```

```
    return 0;
```

```
}
```

- Upotrebimo float ili int promenljivu da bi prikazali broj „pi“ (izbrisati jednu ili drugu //).
- Očigledno je koliko je bitno odabrati adekvatnu promenljivu.
- Promenljiva tipa „double“ je promenljiva „float“ dvostruke preciznosti.

# Tipovi promenljivih - auto

```
#include <iostream>
using namespace std;

int main()
{
    //Demonstracija korišćenje naziva "auto"
    auto coinFlippedHeads = true;
    auto largeNumber = 250000000000;

    cout << "coinFlippedHeads = " << coinFlippedHeads;
    cout << " , sizeof(coinFlippedheads) = " << sizeof(coinFlippedHeads) << endl;

    /*Komanda "sizeof,, se koristi kada želimo da dobijemo informaciju o količini memorije
    koju kompajler određuje za neku promenljivu*/

    cout << "largeNumber = " << largeNumber;
    cout << " , sizeof(largeNumber) = " << sizeof(largeNumber) << endl;

    //cout << pi;

    return 0;
}
```

- Savremeni kompajleri obezbeđuju da ne moramo eksplicitno da identifikujemo tip promenljive. Tada koristimo reč „auto“.
- **Oprez: Neki kompajleri ne podržavaju ovu opciju.**
- \*Komanda "sizeof,, se koristi kada želimo da dobijemo informaciju o količini memorije koju kompajler određuje za neku promenljivu.

```
coinFlippedHeads =1, sizeof(coinFlippedheads)=1
largeNumber = 250000000, sizeof(largeNumber) = 8
```



# Tipovi promenljivih

```
#include <iostream>
using namespace std;

int main()
{
    //Demonstracija posledica razlike u tipu promenljivih

    float pi1 = 3.14159;
    int pi2 = 3.14159; /*ceo broj*/
    bool pi3 = 3.14159; /*0 ili 1*/
    double pi4 = 3.14159; /*dvostruka preciznost u odnosu na float*/
    auto pi5 = 3.14159; /*sam procenjuje*/

    cout << "float pi = " << pi1 << endl;
    cout << "int pi = " << pi2 << endl;
    cout << "bool pi = " << pi3 << endl;
    cout << "double pi = " << pi4 << endl;
    cout << "auto pi = " << pi5 << endl;

    cout << "\n"; /*Novi red*/

    //Komanda "sizeof" se koristi kada želimo da dobijemo informaciju o količini
    memorije (u bajtovima) ... (1 bajt = 8 bitova) ... koju kompajler određuje za neku
    promenljivu

    cout << "sizof float pi = " << sizeof(pi1) << endl;
    cout << "sizof int pi = " << sizeof(pi2) << endl;
    cout << "sizof bool pi = " << sizeof(pi3) << endl;
    cout << "sizof double pi = " << sizeof(pi4) << endl;
    cout << "sizof auto pi = " << sizeof(pi5) << endl;

    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

- Savremeni kompajleri obezbeđuju da ne moramo eksplicitno da identifikujemo tip promenljive. Tada koristimo reč „auto“.
- \*Komanda "sizeof,, se koristi kada želimo da dobijemo informaciju o količini memorije u bajtovima koju kompajler određuje za neku promenljivu.

```
float pi = 3.14159
int pi = 3
bool pi = 1
double pi = 3.14159
auto pi = 3.14159
```

```
sizof float pi = 4
sizof int pi = 4
sizof bool pi = 1
sizof double pi = 8
sizof auto pi = 8
```

Press any key to continue . . .

# Konstante



```
//Konstante
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()  
{
```

```
//Literarne konstante
```

```
cout << "Hello world!" << endl;
```

```
int nekiBroj = 10; cout << nekiBroj << endl;
```

```
//Deklarisane konstante. One se deklarisu pomocu komande "const".
```

```
const double pi = 22.0/7;
```

```
cout << "Vrednost konstante pi je: " << pi << endl;
```

```
/*Probajte da kompajirate program bez komande "double"*/
```

```
//Konstanta koje mogu samo sa sadrze odredjene vrednosti
```

```
return 0;
```

```
}
```

- Literarne konstante mogu biti reči ili brojevi.
- Deklarisane konstante se kreiraju korišenjem komande „const“.

Hello world!

10

Vrednost konstante pi je: 3.14286

# Konstante i prikaz fiksiranog broja decimala



```
//Konstante i prikaz fiksiranog broja decimala
```

```
#include <iostream>
#include <iomanip> /*da bi mogli kasnije da koristimo komandu setprecision*/
using namespace std;
```

```
int main()
{
```

```
//Literarne konstante (mogu biti reci ili brojevi)
```

```
    cout << "Cao studenti!" << endl;
    int nekiBroj = 10; cout << nekiBroj << endl;
    cout << "\n";
```

```
//Deklarisane konstante: One se deklarisu pomocu komande "const"
```

```
    const double pi = 22.0/7.0;
    cout << "Vrednost konstante pi je " << pi;
    cout << " ... ili preciznije ";
    std::cout << std::fixed << std::setprecision(15) << pi << endl; /*Prikaz konstante sa 15
decimala*/
```

```
/*Probajte da kompajlirate program bez komande "double"*/
```

```
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
```

```
}
```

Cao studenti!

10

Vrednost konstante pi je 3.14286 ... ili preciznije  
3.142857142857143

Press any key to continue ...

- Literarne konstante mogu biti reči ili brojevi.
- Deklarisane konstante se kreiraju korišenjem komande „const“.

# Konstante - nabranje



```
#include <iostream>
using namespace std;
/*Ako želimo da definišemo promenljive kojima je dozvoljeno da prihvataju samo određeni set vrednosti koristimo komandu "enum"*/
/*Najpre napravimo promenljivu koja može da ima različite vrednosti*/
enum Ocena
{
    Nedovoljan = 1, /*Da ovde nismo upisali 1, kompajler bi automatski ovoj promenljivoj dodelio vrednost 0*/
    Dovoljan, /*Kompajler, po automatizmu, svakoj sledećoj opcionalnoj promenljivoj dodeljuje za 1 veći broj*/
    Dobar,
    Vrlo_dobar,
    Odlican
};

/*Zapaziti da je definisala gornjeg niza promenljivih napisana pre "main" dela programa*/
int main()
{
    cout << "Nedovoljan je ocena: " << Nedovoljan << endl;
    cout << "Dovoljan je ocena: " << Dovoljan << endl;
    cout << "Dobar je ocena: " << Dobar << endl;
    cout << "Vrlo dobar je ocena: " << Vrlo_dobar << endl;
    cout << "Odlican je ocena: " << Odlican << endl;
    /*Iskoristimo gore definisano nabranje da bi prikazali ocenu na testu*/
    Ocena OcenaNaTestu = Dobar; /*Kreiranje promenljive tipa "Ocena"*/
    cout << "Na testu ste dobili ocenu: " << OcenaNaTestu << endl;

    return 0;
}
```

- Učimo komandu „enum“
- Zapaziti korak promene kod numeracije
- Zadatak: Modifikujte program da bi prikazao fakultetsko ocenjivanje (od 6)

```
Nedovoljan je ocena: 1
Dovoljan je ocena: 2
Dobar je ocena: 3
Vrlo dobar je ocena: 4
Odlican je ocena: 5
Na testu ste dobili ocenu: 3
```

# Konstante - nabranje

```
#include <iostream>
using namespace std;
/*Ako želimo da definišemo promenljive kojima je dozvoljeno da prihvataju samo određeni set vrednosti koristimo komandu
"enum"*/
/*Najpre napravimo promenljivu koja može da ima različite vrednosti*/
enum Ocena
{
    Nedovoljan = 1, /*Da ovde nismo upisali 1, kompajler bi automatski ovoj promenljivoj dodelio vrenost 0*/
    Dovoljan, /*Kompajler, po automatizmu, svakoj sledećoj opcionoj promenljivoj dodeljuje za 1 veći broj*/
    Dobar,
    Vrlo_dobar,
    Odlican
};

/*Zapaziti da je definisane gornjeg niza promenljivih napisana pre "main" dela programa*/
int main()
{
    cout << "Nedovoljan je ocena: " << Nedovoljan << endl;
    cout << "Dovoljan je ocena: " << Dovoljan << endl;
    cout << "Dobar je ocena: " << Dobar << endl;
    cout << "Vrlo dobar je ocena: " << Vrlo_dobar << endl;
    cout << "Odlican je ocena: " << Odlican << endl;
    /*Iskoristimo gore definisano nabranje da bi prikazali ocenu na testu*/
    Ocena OcenaNaTestu = Dobar; /*Kreiranje promenljive tipa "Ocena"*/
    cout << "Na testu ste dobili ocenu: " << OcenaNaTestu << endl;
    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}
```

- Učimo komandu „enum“
- Zapaziti korak promene kod numeracije
- Zadatak: Modifikujte program da bi prikazao fakultetsko ocenjivanje (od 6)

```
Nedovoljan je ocena: 1
Dovoljan je ocena: 2
Dobar je ocena: 3
Vrlo dobar je ocena: 4
Odlican je ocena: 5
Na testu ste dobili ocenu: 3
```

# Program za računanje površine i obima kruga

```
/*Program koji izracunava povrsinu i obim kruga na osnovu unetog poluprecnika*/  
  
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    const double Pi = 3.1416;  
    cout << "Unesite vrednost poluprecnika kruga: ";  
    double radius = 0; /*Pravimo promenljivu "radijus" kojoj omogucujemo da bude double tip  
    promenljive sa inicijalnom vrednoscu 0.*/  
  
    cin >> radius; /*Trazimo unos promenljive "radius"*/  
  
    cout << "Vrednost povrsine kruga je: " << radius*radius*Pi << endl;  
    cout << "Vrednost obima kruga je: " << 2*radius*Pi << endl;  
  
    cin.get(); /*Ovo stavljamo da se exe verzija programa ne bi automatski zatvorila nakon izvršenja programa*/  
    cin.get(); /*Ovo stavljamo da se exe verzija programa ne bi automatski zatvorila nakon izvršenja programa*/  
  
    return 0;  
}
```

- Koristimo znanje za definisanje konstanti
- Zapaziti korak zadavanja inicijalne vrednosti konstante (radijusa)
- Zapaziti poslednje 2 linije koda koji omogućava pauzu na kraju izvršenja programa.

```
Unesite vrednost poluprecnika kruga: 3  
Vrednost povrsine kruga je: 28.2744  
Vrednost obima kruga je: 18.8496
```

# Program za računanje površine i obima kruga

```

/*Program koji izracunava površinu i obim kruga na osnovu unetog poluprecnika*/

#include <iostream>
using namespace std;

int main()
{
    const double Pi = 3.1416;
    cout << "Unesite vrednost poluprecnika kruga: ";
    double radius = 0; /*Pravimo promenljivu "radijus" kojoj omogucujemo da bude double
tip
promenljive sa inicijalnom vrednoscu 0.*/

    cin >> radius; /*Trazimo unos promenljive "radius"*/

    cout << "Vrednost površine kruga je: " << radius * radius * Pi << endl;
    cout << "Vrednost obima kruga je: " << 2 * radius * Pi << endl;

    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}

```

- Koristimo znanje za definisanje konstanti
- Zapaziti korak zadavanja inicijalne vrednosti konstante (radijusa)

```

Unesite vrednost poluprecnika kruga: 3
Vrednost površine kruga je: 28.2744
Vrednost obima kruga je: 18.8496

```

# Nizovi



```
/*Program koji prikazuje zeljeni element u nizu od 5 brojeva*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    cout << "Upisi koji element niza (1,2,3,4,5) zelis da prikazes: ";
```

```
    int brojElementa = 0; /*Kreiranje promenljive, broja, koji cemo koristiti za numeraciju elemenata niza.
```

```
    Zadajemo novoj promenljivoj inicijalnu vrednost 0*/
```

```
    cin >> brojElementa; /*Unos promenljive, broja, za numeraciju elemenata niza*/
```

```
    int elementiNiza [5] = {1,2,3,4,5}; /*Kreiranje niza*/
```

```
    cout << elementiNiza [brojElementa-1]; /* Ispis odredjenog elementa niza. Bitno je primetiti da, posto numeracija niza ide od 0, moramo ovde upisati brojElementa-1, (posto npr. 1 elemant niza ima numeraciju 0)*/
```

```
    cin.get(); /*Ovo stavljamo da se exe verzija programa ne bi automatski zatvorila nakon izvršenja programa*/
```

```
    cin.get(); /*Ovo stavljamo da se exe verzija programa ne bi automatski zatvorila nakon izvršenja programa*/
```

```
    return 0;
```

```
}
```

- Pravimo niz od 5 brojeva
- Zapaziti sintaksu za formiranje niza. U uglastoj zagradi upisujemo broj elemenata niza a same elemente u velikoj zagradi odvojene zapetom.
- Zapamtiti da se po numeraciji, prvom elementu niza automatski dodeljuje broj 0.
- Zato, pri pozivanju elemenata niza, moramo da obratimo paznju da se od trazanog broja oduzme 1.

```
Upisi koji element niza (1,2,3,4,5) zelis da prikazes: 3  
3
```



# Nizovi

```

/*Program koji prikazuje zeljeni element u nizu od 5 brojeva*/

#include <iostream>

using namespace std;

int main()
{
    cout << "Upisi koji elemenat niza (1,2,3,4,5) zelis da prikazes: ";
    int brojElementa = 0; /*Kreiranje promenljive, broja, koji cemo koristiti za numeraciju
elemenata niza.
Zadajemo novoj promenljivoj inicijalnu vrednost 0*/
    cin >> brojElementa; /*Unos promenljive, broja, za numeraciju elemenata niza*/

    int elementiNiza[5] = { 1,2,3,4,5 }; /*Kreiranje niza*/

    cout << elementiNiza[brojElementa - 1]; /* Ispis odredjenog elementa niza. Bitno je primetiti
da, posto
numeracija niza ide od 0, moramo ovde upisati brojElementa-1, (posto npr. 1 elemant niza ima
numeraciju 0)*/

    cout << "\n"; /*Novi red*/
    system("pause"); /*Pauza*/
    return 0;
}

```

- Pravimo niz od 5 brojeva
- Zapaziti sintaksu za formiranje niza. U uglastoj zagradi upisujemo broj elemenata niza a same elemente u velikoj zagradi odvojene zapetom.
- Zapamtiti da se po numeraciji, prvom elementu niza automatski dodjeljuje broj 0.
- Zato, pri pozivanju elemenata niza, moramo da obratimo paznju da se od trazenog broja oduzme 1.

Upisi koji elemenat niza (1,2,3,4,5) zelis da prikazes: 3  
3

# 2D nizovi



```
/*Program koji prikazuje zeljeni element u 2D nizu od 2 reda i 3 kolone brojeva
123
456*/

#include <iostream>

using namespace std;

int main()
{
    cout << "Upisi iz kog reda niza zelis da prikazes broj: ";
    int brojReda = 0;
    cin >> brojReda;

    cout << "Upisi iz koje kolone niza zelis da prikazes broj: ";
    int brojKolone = 0;
    cin >> brojKolone;

    int elementiNiza [2] [3] = {{1,2,3},{4,5,6}}; /*Kreiranje niza 2x3*/

    cout << elementiNiza [brojReda-1] [brojKolone-1];
    /*Bitno je primetiti da, posto opet numeracija niza ide od 0, moramo ovde upisati brojReda-1, i brojKolone-1*/

    cin.get(); /*Ovo stavljamo da se exe verzija programa ne bi automatski zatvorila nakon izvršenja programa*/
    cin.get(); /*Ovo stavljamo da se exe verzija programa ne bi automatski zatvorila nakon izvršenja programa*/

    return 0;
}
```

- Pravimo niz od 2x3 brojeva
- Zapaziti sintaksu za formiranje 2D niza. Slicno sintaksi za 1D niz, u uglastoj zagradi upisujemo brojeve elemenata niza (prvo broj redova pa kolona), a same elemente u velikoj zagradi odvojene zaptom.
- Zapazite da se po numeraciji, prvim elementima redova i kolona niza automatski dodeljuje broj 0.

```
Upisi iz kog reda niza zelis da prikazes broj: 2
Upisi iz koje kolone niza zelis da prikazes broj: 2
5
```

# 2D nizovi

```
/*Program koji prikazuje zeljeni element u 2D nizu od 2 reda i 3 kolone brojeva
```

```
123
```

```
456*/
```

```
#include <iostream>  
using namespace std;
```

```
int main()
```

```
{
```

```
    cout << "Imamo 2D niz: " << "\n";
```

```
    cout << "1 2 3" << "\n";
```

```
    cout << "4 5 6" << "\n";
```

```
    cout << "Upisi iz kog reda niza zelis da prikazes broj: ";
```

```
    int brojReda = 0;
```

```
    cin >> brojReda;
```

```
    cout << "Upisi iz koje kolone niza zelis da prikazes broj: ";
```

```
    int brojKolone = 0;
```

```
    cin >> brojKolone;
```

```
    int elementiNiza[2][3] = { {1,2,3},{4,5,6} }; /*Kreiranje niza 2x3*/
```

```
    cout << elementiNiza[brojReda - 1][brojKolone - 1];
```

```
    /*Bitno je primetiti da, posto opet numeracija niza ide od 0, moramo ovde upisati brojReda-1, i brojKolone-1*/
```

```
    cout << "\n"; /*Novi red*/
```

```
    system("pause"); /*Pauza*/
```

```
    return 0;
```

```
}
```

- Pravimo niz od 2x3 brojeva
- Zapaziti sintaksu za formiranje 2D niza. Slicno sintaksi za 1D niz, u uglastoj zagradi upisujemo brojeve elemenata niza (prvo broj redova pa kolona), a same elemente u velikoj zagradi odvojene zapetom.
- Zapazite da se po numeraciji, prvim elementima redova i kolona niza automatski dodeljuje broj 0.

Imamo 2D niz:

1 2 3

4 5 6

Upisi iz kog reda niza zelis da prikazes broj: 2

Upisi iz koje kolone niza zelis da prikazes broj: 3

6

Press any key to continue . . .