

Programiranje

DRUGO PREDAVANJE

VARIJABLE I TIPOVI PODATAKA

Varijable i tipovi podataka

- ULAZ: Cijeli brojevi A i B
- IZLAZ: $A+B$, $B-A$, $A*B$ i A/B

Varijable i tipovi podataka

- ULAZ: Cijeli brojevi A i B
- IZLAZ: $A+B$, $B-A$, $A*B$ i A/B
 1. Učitaj broj A
 2. Učitaj broj B
 3. Ispiši $A+B$
 4. Ispiši $B-A$
 5. Ispiši $A*B$
 6. Ispiši A/B

Varijable i tipovi podataka

```
#include <iostream>
int main () {
1.  Učitaj broj A
2.  Učitaj broj B
3.  Ispiši A+B
4.  Ispiši B-A
5.  Ispiši A*B
6.  Ispiši A/B
    return 0;
}
```

Osnovni kostur C++
programa

Osnovni kostur C++ programa

Deklaracijski dio

Sastoji se od pretprocesorskih i deklarativnih naredbi koje daju osnovne definicije biblioteka, konstanti i drugih elemenata bitnih za rad programa.

Izvedbeni dio

Sastoji se od izvršnog dijela programa kojeg tvore funkcije. Među njima postoji jedna istaknuta funkcija koja nosi naziv **main**.

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
1.  Učitaj broj A
```

```
2.  Učitaj broj B
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Deklaracijski dio

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
1.  Učitaj broj A
```

```
2.  Učitaj broj B
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Pretprocesorska naredba za uključivanje biblioteke funkcija u program.

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
1.  Učitaj broj A
```

```
2.  Učitaj broj B
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Biblioteka funkcija za rad sa standardnim ulazno-izlaznim jedinicama (zaslona i tipkovnica).

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
1.  Učitaj broj A
```

```
2.  Učitaj broj B
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Izvedbeni dio

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
1.  Učitaj broj A
```

```
2.  Učitaj broj B
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Početak i kraj **main** funkcije.

Varijable i tipovi podataka

```
#include <iostream>
int main () {
1.  Učitaj broj A
2.  Učitaj broj B
3.  Ispiši A+B
4.  Ispiši B-A
5.  Ispiši A*B
6.  Ispiši A/B
    return 0;
}
```

Vitičaste zagrade definiraju **blok naredbi**. Naredbe u bloku se izvode redom kojem su navedene. Zbog toga one čine **slijed** ili **sekvenciju**.

Varijable i tipovi podataka

```
#include <iostream>
int main () {
1.  Učitaj broj A
2.  Učitaj broj B
3.  Ispiši A+B
4.  Ispiši B-A
5.  Ispiši A*B
6.  Ispiši A/B
    return 0;
}
```

Poruka OS-u da je program korektno završio.

Varijable i tipovi podataka

```
#include <iostream>
int main () {
1.  Učitaj broj A
2.  Učitaj broj B
3.  Ispiši A+B
4.  Ispiši B-A
5.  Ispiši A*B
6.  Ispiši A/B
    return 0;
}
```

Svaka naredba u programskom jeziku C++ završava znakom ';' koji se naziva **delimiter**.

Varijable i tipovi podataka

```
#include <iostream>
int main () {
1.  Učitaj broj A
2.  Učitaj broj B
3.  Ispiši A+B
4.  Ispiši B-A
5.  Ispiši A*B
6.  Ispiši A/B
    return 0;
}
```

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A;
```

```
    int B;
```

```
1.    Učitaj broj A
```

```
2.    Učitaj broj B
```

```
3.    Ispiši A+B
```

```
4.    Ispiši B-A
```

```
5.    Ispiši A*B
```

```
6.    Ispiši A/B
```

```
    return 0;
```

```
}
```

Deklaracija varijabli A i B.

Varijable i tipovi podataka

- ❑ **Varijabla** je imenovani dio memorije koji sadrži podatak koji se koristi u programu.
- ❑ Varijabli se može dodijeliti bilo koja vrijednost u domeni tipa podataka varijable.
- ❑ Vrijednost varijable se tijekom rada programa može mijenjati (za razliku od primjerice u matematici).

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A;
```

```
    int B;
```

1. Učitaj broj A

2. Učitaj broj B

3. Ispiši A+B

4. Ispiši B-A

5. Ispiši A*B

6. Ispiši A/B

```
    return 0;
```

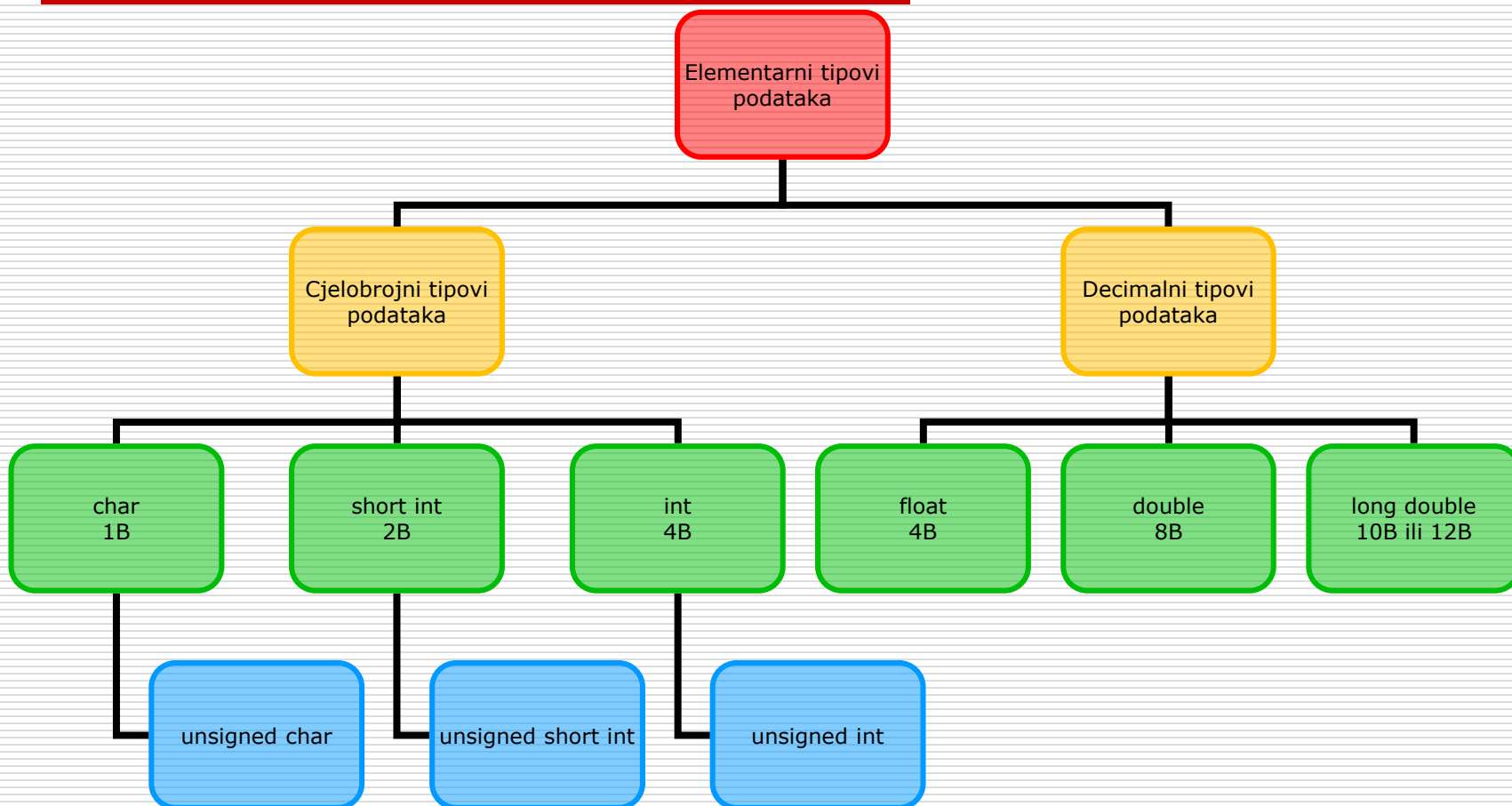
```
}
```

Tip podataka za varijable

Tipovi podataka u C++

- **Tip podataka** definira:
 - Količinu memorijskog prostora koji će varijabla zauzimati
 - Kako će vrijednost biti zapisana u memoriji
 - Koje će se operacije moći izvesti nad podatkom
- Svi osnovni ili elementarni tipovi podataka u programskom jeziku C++ su brojevi:
 - Cijeli brojevi
 - Decimalni brojevi u notaciji pomičnog zareza

Tipovi podataka u C++



Cjelobrojni tipovi podataka

- **char**
- Veličina: 1B
- Raspon: od -128 do 127
- **Napomena:** Kod unosa i ispisa za tip podataka **char** programski jezik ima internu transformaciju između brojeva i znakova prema ASCII (eng. American Standard Code for Information Interchange) tablici

Cjelobrojni tipovi podataka

- **short int** (sinonimi: short)
- Veličina: 2B
- Raspon: od -32768 do 32767

Cjelobrojni tipovi podataka

- **int** (sinonimi: long int, long)
- Veličina: 4B
- Raspon: od - 2147483648 do 2147483647

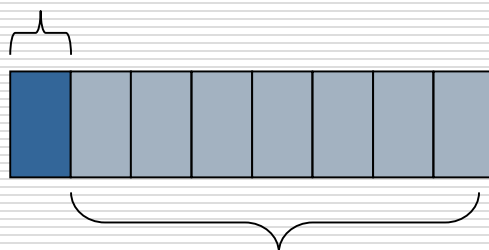
Cjelobrojni tipovi podataka

- Zašto negativnih brojeva u svakom tipu podataka ima za jedan više nego pozitivnih?

Cjelobrojni tipovi podataka

- Način reprezentacije cijelih brojeva
- **Primjer:** tip char

predznak



binarna
reprezentacija
broja

Cjelobrojni tipovi podataka

- Ako bi se jednostavno uzelo da je prvi bit predznak, a preostalih 7 bitova iznos broja, brojevi:

00000000 i

10000000

bi oba predstavljali broj 0.

Cjelobrojni tipovi podataka

- Zato se koristi notacija tzv. dualnog komplementa u kojem negativni broj k ima značenje $\sim k + 1$. Tako primjerice imamo:

$$\begin{aligned} 10001011 &= \sim 10001011 + 1 \\ &= -(01110100 + 1) \\ &= -01110101 \\ &= -117 \end{aligned}$$

Cjelobrojni tipovi podataka

- ❑ Ponekad nam negativni brojevi nisu potrebni.
- ❑ Ispred bilo kojeg cjelobrojnog tipa podataka može se staviti ključna riječ **unsigned** i onda se vodeći bit broja neće koristiti za predznak, već će se svi bitovi koristiti za iznos broja

Cjelobrojni tipovi podataka

- Stoga imamo sljedeće raspone:
 - unsigned char – od 0 do 255
 - unsigned short – od 0 do 65535
 - unsigned int – od 0 do 4294967295

Decimalni tipovi podataka

□ float

□ Veličina: 4B

□ Raspon: od $-3.4 \cdot 10^{38}$ do $-3.4 \cdot 10^{-38}$,
od $3.4 \cdot 10^{-38}$ do $3.4 \cdot 10^{38}$.

□ Zajamčena točnost: 7 decimala

Decimalni tipovi podataka

□ double

□ Veličina: 8B

□ Raspon: od $-1.7 \cdot 10^{308}$ do $-1.7 \cdot 10^{-308}$
od $1.7 \cdot 10^{-308}$ do $1.7 \cdot 10^{308}$

□ Zajamčena točnost: 15 decimala

Decimalni tipovi podataka

- **long double**
- Tip podataka long double još nije standardiziran i njegova duljina ovisi o prevoditelju.
- Ne preporuča se njegovo korištenje ukoliko za to ne postoji prijeka potreba.

Decimalni tipovi podataka

- Eksponencijalna ili znanstvena notacija
 - $370000. = 3.7e5 (3.7 \cdot 10^5)$
 - $0.00012 = 1.2e-4 (1 \cdot 10^{-4})$
- Kod reprezentacije broja posebno se reprezentira **mantisa**, a posebno **eksponent**.
- I mantisa i eksponent imaju predznak.

Decimalni tipovi podataka

- ❑ **Oprez 1:** Brojevi zapisani u ovoj notaciji najgušći su oko 0, a što je udaljenost od 0 veća, sve su rjeđi.
- ❑ **Oprez 2:** Kod decimalnih se brojeva ne koristi notacija dualnog komplementa.
- ❑ **Oprez 3:** U programskom jeziku C++ ne postoje **unsigned** decimalni brojevi.

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A;
```

```
    int B;
```

1. Učitaj broj A

2. Učitaj broj B

3. Ispiši A+B

4. Ispiši B-A

5. Ispiši A*B

6. Ispiši A/B

```
    return 0;
```

```
}
```

Imena ili identifikatori
varijabli

Identifikatori u C++

- Identifikatori se mogu sastojati od:
 - Slova (velikih i malih)
 - Brojeva
 - Specijalnog znaka _

- Dodatno ograničenje:
 - Prvi znak identifikatora ne može biti broj

- **Oprez 1:** C++ tretira velika i mala slova kao različite znakove, pa su varijable a i A različite varijable.
- **Oprez 2:** Znakovi č, ć, đ, š, ž, Č,Ć, Đ, Š, Ž smatraju se specijalnim znakovima i **ne mogu** se koristiti u identifikatorima.

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A;
```

```
    int B;
```

1. Učitaj broj A

2. Učitaj broj B

3. Ispiši A+B

4. Ispiši B-A

5. Ispiši A*B

6. Ispiši A/B

```
    return 0;
```

```
}
```

Više varijabli istog tipa podataka moguće je deklarirati zajedno.

Varijable i tipovi podataka

```
#include <iostream>
int main () {
    int A, B;
    1.  Učitaj broj A
    2.  Učitaj broj B
    3.  Ispiši A+B
    4.  Ispiši B-A
    5.  Ispiši A*B
    6.  Ispiši A/B
    return 0;
}
```

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A, B;
```

```
    std::cin >> A;
```

```
    std::cin >> B;
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

cin je ulazni tok podataka koji predstavlja standardnu ulaznu jedinicu (tipkovnicu)

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A, B;
```

```
    std::cin >> A;
```

```
    std::cin >> B;
```

3. Ispiši A+B

4. Ispiši B-A

5. Ispiši A*B

6. Ispiši A/B

```
    return 0;
```

```
}
```

Svi ulazno-izlazni tokovi u C++ su definirani u prostoru funkcijskih imena (**namespace**) **std**, pa se on mora navesti kako bi se tokovi u njemu opisani mogli upotrebljavati

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A, B;
```

```
    std::cin >> A;
```

```
    std::cin >> B;
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Operator `::` se naziva
operator dodjele područja ili
operator pripadnosti, a koristi se za
pristup prostoru funkcijskih imena
(namespace) `std`

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A, B;
```

```
    std::cin >> A;
```

```
    std::cin >> B;
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Operator za čitanje (unos)
podataka sa ulaznog toka
podataka.

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A, B;
```

```
    std::cin >> A;
```

```
    std::cin >> B;
```

```
3.  Ispiši A+B
```

```
4.  Ispiši B-A
```

```
5.  Ispiši A*B
```

```
6.  Ispiši A/B
```

```
    return 0;
```

```
}
```

Operator `>>` nije standardni C++ operator, već je dio biblioteke `iostream`.

Varijable i tipovi podataka

```
#include <iostream>
int main () {
    int A, B;
    std::cin >> A;
    std::cin >> B;
    std::cout << A+B;
    std::cout << B-A;
    std::cout << A*B;
    std::cout << A/B;
    return 0;
}
```

Standardni izlazni tok podataka koji definira standardnu izlaznu jedinicu (zaslon).

Varijable i tipovi podataka

```
#include <iostream>
int main () {
    int A, B;
    std::cin >> A;
    std::cin >> B;
    std::cout << A+B;
    std::cout << B-A;
    std::cout << A*B;
    std::cout << A/B;
    return 0;
}
```

Operator za ispis podataka
na izlazni tok podataka.

Varijable i tipovi podataka

```
#include <iostream>
```

```
int main () {
```

```
    int A, B;
```

```
    std::cin >> A;
```

```
    std::cin >> B;
```

```
    std::cout << A+B;
```

```
    std::cout << B-A;
```

```
    std::cout << A*B;
```

```
    std::cout << A/B;
```

```
    return 0;
```

```
}
```

Aritmetički operatori

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cin >> A;
    cin >> B;
    cout << A+B;
    cout << B-A;
    cout << A*B;
    cout << A/B;
    return 0;
}
```

Kako se ne bi trebalo ispred svakog toka podataka iz prostora imena pisati identifikator prostora, on se može definirati posebnom naredbom u deklaracijskom dijelu programa.

Varijable i tipovi podataka

- ❑ Kad pokrenemo ovaj program, na kratko će se pojaviti prozor i nestati.
- ❑ Problem je u tome što MS Windows zatvori prozor odmah nakon što program završi.

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cin >> A;
    cin >> B;
    cout << A+B;
    cout << B-A;
    cout << A*B;
    cout << A/B;
    system("pause");
    return 0;
}
```

Funkcija **system** je namijenjena pokretanju naredbe OS MS Windows. Dakle, ovo neće raditi na ostalim operacijskim sustavima.

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cin >> A;
    cin >> B;
    cout << A+B;
    cout << B-A;
    cout << A*B;
    cout << A/B;
    system("pause");
    return 0;
}
```

```
2
3
5160
Press any key to continue...
```

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cin >> A;
    cin >> B;
    cout << A+B;
    cout << endl;
    cout << B-A;
    cout << A*B;
    cout << A/B;
    system("pause");
    return 0;
}
```

endl je predefinirana konstanta koja ispisuje znak s ASCII kodom 13 (prelazak u novi red)

```
2
3
5
160
Press any key to continue...
```

Varijable i tipovi podataka

- Ako imamo više ulaznih ili izlaznih tokova podataka za redom, možemo ih zapisivati jedne za drugom u istoj naredbi.

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cin >> A;
    cin >> B;
    cout << A+B << endl;
    cout << B-A << endl;
    cout << A*B << endl;
    cout << A/B << endl;
    system("pause");
    return 0;
}
```

```
2
3
5
1
6
0
Press any key to continue...
```

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cout << "Upišite A: ";
    cin >> A;
    cout << "Upišite B: ";
    cin >> B;
    cout << A+B << endl;
    cout << B-A << endl;
    cout << A*B << endl;
    cout << A/B << endl;
    system("pause");
    return 0;
}
```

```
Upišite A: 2
Upišite B: 3
5
1
6
0
Press any key to continue...
```

?

Varijable i tipovi podataka

- Aritmetički operatori vraćaju rezultat koji je onog tipa podataka kojeg su operandi, odnosno općenitijeg od tipova operanada, ako su operandi različitog tipa.
- Kako su oba operanda tipa `int` i rezultat operacije ima tip `int`, dakle dobili smo rezultat cjelobrojnog dijeljenja

Varijable i tipovi podataka

Operandi	Rezultat
short char	short
int char	int
int short	int
unsigned int int	int
float char	float
double float	double

Varijable i tipovi podataka

- Kako bismo pretvorili rezultat u decimalni broj, moramo barem jedan operand pretvoriti u decimalni broj.

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cout << "Upišite A: ";
    cin >> A;
    cout << "Upišite B: ";
    cin >> B;
    cout << A+B << endl;
    cout << B-A << endl;
    cout << A*B << endl;
    cout << (double)A/B
        << endl;
    system("pause");
    return 0;
}
```

Naredba se može protezati kroz više redaka.

```
Upišite A: 2
Upišite B: 3
5
1
6
0.666667
Press any key to continue...
```

Operand A pretvorili smo u podatak tipa **double**, pa će i rezultat biti tipa **double**.

Varijable i tipovi podataka

- Može li se, kad već imamo relativno jednostavno izvedivo cjelobrojno dijeljenje, jednostavno izračunati i ostatak od dijeljenja?

Varijable i tipovi podataka

```
#include <iostream>
using namespace std;
int main () {
    int A, B;
    cout << "Upišite A: ";
    cin >> A;
    cout << "Upišite B: ";
    cin >> B;
    cout << A+B << endl;
    cout << B-A << endl;
    cout << A*B << endl;
    cout << A/B;
    cout << " i " << A%B
        << " ostatka " << endl;
    system("pause");
    return 0;
}
```

Ispis definiranog teksta.

Ostatak od dijeljenja.

```
Upišite A: 2
Upišite B: 3
5
1
6
0 i 2 ostatka
Press any key to continue...
```

Primjer 2 – prebacivanje u razne sustave

- Ulaz: Pozitivni cijeli broj A
- Izlaz: Broj A zapisan u heksadecimalnom i oktalnom sustavu

Primjer 2 – prebacivanje u razne sustave

□ Algoritam:

1. Ispiši "A="
2. Učitaj cijeli broj A
3. Ako je $A \leq 0$ vrati se na 1
4. Prebaci format ispisa cijelih brojeva u oktalni sustav
5. Ispiši broj A
6. Prebaci format ispisa cijelih brojeva u heksadecimalni sustav
7. Ispiši broj A

Primjer 2 – prebacivanje u razne sustave

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

Deklaracijski dio programa

```
int main () {
```

- Ispiši "A="
 - Učitaj cijeli broj A
 - Prebaci format ispisa cijelih brojeva u oktalni sustav
 - Ispiši broj A
 - Prebaci format ispisa cijelih brojeva u heksadecimalni sustav
 - Ispiši broj A
- ```
return 0; }
```

Izvršni  
dio  
programa

# Primjer 2 – prebacivanje u razne sustave

---

- ❑ **#include** – naredba za uključivanje biblioteke funkcija u program
  - ❑ `iostream` – biblioteka standardnih U/I funkcija
  - ❑ **using namespace** – naredba za definiranje prostora funkcijskih imena koji se koristi u programu
  - ❑ `std` – prostor funkcijskih imena koji sadrži imena standardnih U/I tokova
  - ❑ `int main () {` - početak definicije glavne funkcije programa
  - ❑ **return 0** – dolazi na kraju programa kako bi se OS-u vratila poruka da je program korektno završio
  - ❑ `{ }` – blok programskih naredbi koje se izvršavaju slijedno - **sekvencija**
-

# Primjer 2 – prebacivanje u razne sustave

---

```
#include <iostream>
using namespace std;
int main () {
 unsigned short A;
 cout << "A=";
 cin >> A;
 3. Prebaci format ispisa cijelih brojeva u oktalni sustav
 4. Ispiši broj A
 5. Prebaci format ispisa cijelih brojeva u heksadecimalni sustav
 6. Ispiši broj A
 return 0;
}
```



# Primjer 2 – prebacivanje u razne sustave

---

- ❑ `unsigned short` – dvobajtni cjelobrojni tip podataka koji sadrži samo nenegativne brojeve
  - ❑ `A` – identifikator varijable
    - Identifikatori se kreiraju od
      - ❑ Slova (malih i velikih)
      - ❑ Brojki
      - ❑ Znaka `'_'`
      - ❑ Prvi znak ne smije biti brojka
      - ❑ Velika i mala slova se razlikuju
    - `cout` – standardni izlazni tok podataka
    - `<<` – naredba za ispis podataka na izlazni tok podataka
    - `" "` – Unutar dvostrukih navodnika navodi se tekst koji želimo ispisati u `cout` naredbi
    - `cin` – standardni ulazni tok podataka
    - `>>` – naredba za učitavanje podataka sa ulaznog toka podataka
-

# Primjer 2 – prebacivanje u razne sustave

---

```
#include <iostream>
```

```
using namespace std;
```

```
int main () {
```

```
 unsigned short A;
```

```
 cout << "A=";
```

```
 cin >> A;
```

```
 cout << oct << A << endl;
```

5. Prebaci format ispisa cijelih brojeva u heksadecimalni sustav

6. Ispiši broj A

```
 return 0;
```

```
}
```

# Primjer 2 – prebacivanje u razne sustave

---

- ❑ `oct` – instrukcija koja sve ispise cijelih brojeva prebacuje u oktalni zapis. Nakon nje će `cout` naredbe koje slijede sve cijele brojeve ispisivati u oktalnom zapisu sve dok se ova opcija ne isključi
- ❑ `endl` – konstanta čija je vrijednost postavljena na `"\n"`. To znači da ona sadrži broj 13, koji predstavlja znak za prelazak u novi red
- ❑ Za jedan izlazni tok podataka `cout` naredbe može se navesti više ispisa koji su međusobno odvojeni operatorima `<<`

# Primjer 2 – prebacivanje u razne sustave

---

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

int main () {
 unsigned short A;
 cout << "A=";
 cin >> A;
 cout << oct << A << endl;
 cout << hex << A << endl;
 return 0;
}
```

# Primjer 2 – prebacivanje u razne sustave

---

- ❑ `hex` – instrukcija koja sve ispise cijelih brojeva prebacuje u heksadecimalni zapis. Nakon nje će `cout` naredbe koje slijede sve cijele brojeve ispisivati u heksadecimalnom zapisu sve dok se ova opcija ne isključi
- ❑ Ovako napravljen program će se izvršiti i prozor u kojem se izvršava odmah će se zatvoriti.
- ❑ Da bi to izbjegli, koristit ćemo naredbu OS-a `PAUSE`, koja zaustavlja rad sve dok se ne pritisne neka tipka.

# Primjer 2 – prebacivanje u razne sustave

---

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main () {
 unsigned short A;
 cout << "A=";
 cin >> A;
 cout << oct << A << endl;
 cout << hex << A << endl;
 system("pause");
 return 0;
}
```

# Primjer 2 – prebacivanje u razne sustave

---

- ❑ `system` – funkcija pomoću koje se izvršava naredba OS-a
- ❑ Funkcija `system` je dio biblioteke funkcija s nazivom `cstdlib`
- ❑ **Oprez!** Ovo rješenje radi pod MS Windows operacijskim sustavom, ali ne i pod Linux operacijskim sustavom.

# Primjer 3 – korištenje decimalnih brojeva

---

- Ulaz: Tri decimalna broja  $A$ ,  $B$  i  $C$
- Izlaz:  $(A+B)/C$



# Primjer 3 – korištenje decimalnih brojeva

---

- Ulaz: Tri decimalna broja  $A$ ,  $B$  i  $C$
- Izlaz:  $(A+B)/C$
- 1. Ispiši "A="
- 2. Učitaj decimalni broj  $A$
- 3. Ispiši "B="
- 4. Učitaj decimalni broj  $B$
- 5. Ispiši "C="
- 6. Učitaj decimalni broj  $C$
- 7.  $Z = A + B$
- 8. Ispiši  $Z / C$

# Primjer 3 – korištenje decimalnih brojeva

---

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main () {
1. Ispiši "A="
2. Učitaj decimalni broj A
3. Ispiši "B="
4. Učitaj decimalni broj B
5. Ispiši "C="
6. Učitaj decimalni broj C
7. $Z = A + B$
8. Ispiši Z / C
 system("pause");
 return 0;
}
```

# Primjer 3 – korištenje decimalnih brojeva

---

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main () {
 double A, B, C;
 1. Ispiši "A="
 2. Učitaj decimalni broj A
 3. Ispiši "B="
 4. Učitaj decimalni broj B
 5. Ispiši "C="
 6. Učitaj decimalni broj C
 7. $Z = A + B$
 8. Ispiši Z / C
 system("pause");
 return 0; }
```

# Primjer 3 – korištenje decimalnih brojeva

---

- ❑ `double` – 8-bajtni decimalni broj
- ❑ Unutar iste deklaracije može se deklarirati više varijabli istog tipa podataka i one slijede iza naziva tipa podataka i odvojene su zarezom

# Primjer 3 – korištenje decimalnih brojeva

---

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main () {
 double A, B, C;
 cout << "A=";
 cin >> A;
 cout << "B=";
 cin >> B;
 cout << "C=";
 cin >> C;
 6. Z = A + B
 7. Ispiši Z / C
 system("pause");
 return 0; }
```

# Primjer 3 – korištenje decimalnih brojeva

---

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main () {
 double A, B, C;
 cout << "A=";
 cin >> A;
 cout << "B=";
 cin >> B;
 cout << "C=";
 cin >> C;
 double Z;
 Z = A + B;
 7. Ispiši Z / C
 system("pause");
 return 0;}
```

# Primjer 3 – korištenje decimalnih brojeva

---

- Varijabla  $z$  je međurezultat
- Za razliku od drugih jezika, u programskom jeziku C++ varijable se deklariraju u trenutku kada nam zatrebaju
- **Važno!** Svaku varijablu treba deklarirati prije njena prvog korištenja

# Primjer 3 – korištenje decimalnih brojeva

---

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main () {
 double A, B, C;
 cout << "A=";
 cin >> A;
 cout << "B=";
 cin >> B;
 cout << "C=";
 cin >> C;
 double Z;
 Z = A + B;
 cout << Z / C << endl;
 system("pause");
 return 0;}
```

```
A=17
B=25
C=47
0.893617
Press any key to continue . . .
```



# Primjer 3 – korištenje decimalnih brojeva

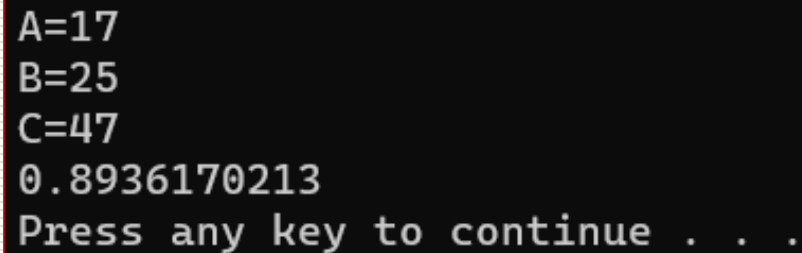
---

- ☐ Tip podataka `double` ima točnost na 15 decimala, no ipak se ispisuje samo 6 decimala
- ☐ Da li je moguće ispisati više od 6 decimala?

# Primjer 3 – korištenje decimalnih brojeva

---

```
#include <iostream>
#include <cstdlib>
#include <iomanip>
using namespace std;
int main () {
 double A, B, C;
 cout << "A=";
 cin >> A;
 cout << "B=";
 cin >> B;
 cout << "C=";
 cin >> C;
 double Z;
 Z = A + B;
 cout << setprecision (10) << Z / C << endl;
 system("pause");
 return 0;}
```

A screenshot of a terminal window showing the output of the program. The text is as follows:  
A=17  
B=25  
C=47  
0.8936170213  
Press any key to continue . . .

# Primjer 3 – korištenje decimalnih brojeva

---

- ❑ `iomanip` – biblioteka koja sadrži funkcije za formatiranje ispisa
- ❑ `setprecision` – funkcija iz biblioteke `iomanip` koja se koristi unutar `cout` naredbe i definira broj znamenaka koji se prikazuje kod prikaza decimalnih brojeva





**KEEP  
CALM!**

**TO BE  
CONTINUED...**