

Vežba 2

Realni brojevi

Zadatak 1. Napisati program koji izračunava porez na nabavnu cenu i maloprodajnu cenu proizvoda. Stopa poreza je konstantna, dok se nabavna cena unosi sa tastature.

Rezultate prikazati u formatiranom ispisu. Za promenljive koristiti realne brojeve tipa **double**.

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

int main() {

    // Deklarisanje promenljivih
    const double STOPA_POREZA = 0.2;
    double nabavna_cena, // Nabavna cena proizvoda
           porez, // Porez na posmatrani proizvod
           cena; // Prodajna cena (nabavna_cena + porez)

    // Podesavanje izlaznog niza za prikazivanje iznosa
    cout << setprecision(2);
         << setiosflags(ios::fixed)
         << setiosflags(ios::showpoint);

    // Unos podataka
    cout << "Unesite nabavnu cenu proizvoda : ";
    cin >> nabavna_cena;

    // Izracunavanja
    porez = nabavna_cena * STOPA_POREZA;
    cena = nabavna_cena + porez;

    // Ispisivanje rezultata
    cout << endl;
    cout << "Nabavna cena je : " << setw(11) << nabavna_cena << endl;
    cout << "Porez je : " << setw(18) << porez << endl;
    cout << "Prodajna cena je : " << setw(10) << cena << endl << endl;

    return 0;
}
```

Zadatak 2. Napisati program za kasu u restoranu koji izračunava maloprodajnu cenu obroka, kao i kusur koji je potrebno vratiti gostu restorana na osnovu cene obroka i datog iznosa.

Izvršiti ispisivanje pozdravne i završne poruke. Za promenljive koristiti realne brojeve tipa **double**.

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

int main() {

    // Deklarisanje promenljivih
    const double STOPA_MALOPRODAJNOG_POREZA = 0.2;
```

```

double cena_obroka,      // Nabavna cena obroka za restoran
      iznos_poreza,      // Iznos poreza
      ukupno,           // Ukupno za naplatu
      dati_iznos,       // Iznos koji je dala musterija
      kusur;           // Kusur: dati_iznos - ukupno

// Podesavanje izlaznog formata za ispisivanja iznosa
cout << setprecision(2)
      << setiosflags(ios::fixed)
      << setiosflags(ios::showpoint);

// Ispisivanje naziva restorana i unos cene obroka
cout << "***** Motel Era *****" << endl << endl;
cout << "Ibarska magistrala b.b." << endl << endl;
cout << "Unesite cenu obroka : EUR ";
cin >> cena_obroka;
cout << endl;

// Izracunavanje poreza i ukupne cene
iznos_poreza = cena_obroka * STOPA_MALOPRODAJNOG_POREZA;
ukupno = cena_obroka + iznos_poreza;

// Ispisivanje poreza i ukupne cene

cout << endl;
cout << "Cena obroka : " << setw(7) << cena_obroka << endl;
cout << "Porez : " << setw(13) << iznos_poreza << endl;
cout << "-----" << endl;
cout << "Ukupan iznos : " << setw(6) << ukupno << endl;

// Unos datog iznosa
cout << endl;
cout << "Unesite iznos koji je musterija dala : EUR ";
cin >> dati_iznos;
cout << endl;

// Izracunavanje kusura
kusur = dati_iznos - ukupno;

// Ispisivanje kusura
cout << endl;
cout << "Dati iznos :   EUR " << setw(9) << dati_iznos << endl;
cout << "Ukupan iznos : EUR " << setw(9) << ukupno << endl;
cout << "-----" << endl;
cout << "Kusur :         EUR " << setw(9) << kusur << endl;

// Ispisivanje završne poruke
cout << endl << endl;
cout << "*** Zivi bili i dosli nam opet ! ***" << endl << endl;

return 0;
}

```

Zadatak 3. Napisati program koji konvertuje dužine unete u miljama i stopama u dužine izražene u kilometrima i metrima. U programu se demonstrira kombinovano korišćenje promenljivih različitog tipa.

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

int main() {

    // Deklarisanje promenljivih
    const double METARA_PO_MILJI = 1609.35;
    const double METARA_PO_STOPI = 0.30480;
    int milje, stope, kilometri, metri;
    double ukupno_metri, ukupno_kilometri;

    // Unos podataka
    cout << "Unesite broj milja : ";
    cin >> milje;
    cout << "Unesite broj stopa : ";
    cin >> stope;

    // pretvaranje unete duzine u metre
    ukupno_metri = milje * METARA_PO_MILJI + stope * METARA_PO_STOPI;

    // Izracunavanje broja kilometara
    ukupno_kilometri = ukupno_metri / 1000;
    kilometri = ukupno_kilometri; // odbacivanje decimalnog dela i
    // dobijanje celog broja kilometara

    // Preracunavanje decimalnog dela kilometara u metre
    metri = (ukupno_kilometri - kilometri) * 1000;

    // Ispisivanje rezultata
    cout << endl;
    cout << "Rastojanje je " << kilometri << " km i "
    << metri << " m" << endl << endl;

    return 0;
}
```

Zadatak 4. U narednom programu se demonstrira upotreba složenih binarnih operatora i to +=, -=, *= i /=.

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

int main(){

    // Deklarisanje promenljivih
    double stanje = 545.50,
           ulog = 300.00,
           podizanje = 200.00,
           broj = 7.5,
           cena = 104.50;

    // Podesavanje izlaznog formata za ispisivanja iznosa
    cout << setprecision(2)
```

```

    << setiosflags(ios::fixed)
    << setiosflags(ios::showpoint);

// Izracunavanja i ispisivanje rezultata na ekranu
stanje += ulog;
cout << "Stanje je " << stanje << endl;

stanje -= podizanje;
cout << "Stanje je " << stanje << endl;

broj *= broj;
cout << "Broj je " << broj << endl;

cena /= 2.0;
cout << "Cena je " << cena << endl;

return 0;
}

```

Zadatak 5. Naredni program ilustruje upotrebu unarnih operatora ispred i iza promenljive.

Upotreba operatora kao u sledećem primeru je veoma česta u C++ jeziku. Zato je neophodno dobro razumeti sledeći primer.

```

#include <iostream>
using namespace std;

int main() {

    // Deklarisanje promenljivih
    int i, j;

    // Izracunavanja i ispisivanje rezultata
    i = 7; // Dodeljivanje inicijalne vrednosti
    j = 4 + --i; // prvo se od i oduzme 1 a zatim se sabere sa 4

    cout << "Pri koriscenju operatora pre promenljive i = "
         << i << " dok je j = " << j << endl;

    i = 7; // Ponovna inicijalizacija promenljive i na 7
    j = 4 + i--; // prvo se i sabere sa 4 a zatim mu se oduzme 1

    cout << "Pri koriscenju operatora posle promenljive i = "
         << i << " dok je j = " << j << endl;

    return 0;
}

```

Iteracije

Zadatak 6. Napisati program koji broji znakove unete sa konzole u jednoj liniji. U program koristiti **while** petlju. U prikazanoj **while** petlji, uslov koji mora biti ispunjen da bi se petlja izvršavala, nalazi se na početku petlje. Ukoliko uslov nije ispunjen, sekvenca naredbi u okviru petlje se neće ni jednom izvršiti.

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

int main() {

    // Deklarisanje promenljivih
    char znak;           // Koristi se za prihvatanje unetog znaka
    int broj_znakova = 0; // Broji unete znakove

    // Unos niza
    cout << "Unesite bilo koji broj znakova i na kraju pritisnite "<< "'Enter'"
         << endl;

    znak = cin.get();    // Unos prvog znaka

    // Petlja za sabiranje znakova unetih u jednom redu
    while (znak != '\n') // While uslov se nalazi na pocetku petlje
    {
        ++broj_znakova;
        znak = cin.get(); // Unos sledeceg znaka
    }

    // Ispisivanje rezultata
    cout << endl;
    cout << "Uneli ste " << broj_znakova << " znakova" << endl;

    return 0;
}
```

Zadatak 7. Izmeniti program iz zadatka 2, tako da može da obradi proizvoljan broj mušterija upotrebom **while** petlje. Kod za izračunavanje se smešta u okviru **while** petlje tako da se može ponavljati željeni broj puta.

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

int main() {

    // Deklarisanje promenljivih
    const double STOPA_MALOPRODAJNOG_POREZA = 0.2;
    double cena_obroka, // Nabavna cena obroka za restoran
           iznos_poreza, // Iznos poreza
           ukupno,      // Ukupno za naplatu
           dati_iznos,  // Iznos koji je dala musterija
           kusur,       // Kusur: dati_iznos - ukupno
           ukupan_promet; // Ukupan promet za posmatrani period

    int broj_musterija, // Sabira ukupan broj obradjenih musterija
        odgovor;       // da = 1, ne = 0
```

```

//Podesavanje izlaznog formata za ispisivanja iznosa
cout << setprecision(2)
      << setiosflags(ios::fixed)
      << setiosflags(ios::showpoint);

// Inicijalizacija brojaca i zbir
ukupan_promet = 0.0;
broj_musterija = 0;

// Ispisivanje naziva restorana
cout << "***** Motel ERA *****" << endl << endl;
cout << "Ibarska magistrala b.b." <<endl << endl;

// Ispitivanje da li se nastavlja unos sledece musterije
cout << endl;
cout << "Da li ima jos neko ko hoce da plati ?" << endl;
cout << "Ako ima kucaj 1 a ako nema kucaj 0 : ";
cin >> odgovor;

// Glavna petlja
while (odgovor == 1) {          // While uslov se nalazi na pocetku petlje
    // Unos cene obroka
    cout << endl;
    cout << "Za koliko para su pojeli i popili : EUR ";
    cin >> cena_obroka;
    cout << endl << endl;

    // Izracunavanje poreza i ukupne cene
    iznos_poreza = cena_obroka * STOPA_MALOPRODAJNOG_POREZA;
    ukupno = cena_obroka + iznos_poreza;

    // Ispisivanje poreza i ukupne cene
    cout << "Pojeli i popili : " << setw(7) << cena_obroka <<endl;
    cout << "Porez : " << setw(17) << iznos_poreza << endl;
    cout << "-----" << endl;
    cout << "Ukupan iznos : " << setw(10) << ukupno << endl << endl;

    // Unos datog iznosa
    cout << "Unesite iznos koji je musterija dala : EUR ";
    cin >> dati_iznos;
    cout << endl << endl;

    // Izracunavanje kusura
    kusur = dati_iznos - ukupno;

    // Ispisivanje kusura
    cout << endl;
    cout << "Dati iznos :   EUR " << setw(9) << dati_iznos << endl;
    cout << "Ukupan iznos : EUR " << setw(9) << ukupno << endl;
    cout << "-----" << endl;
    cout << "Kusur :           EUR " << setw(9) << kusur << endl;

    // Povecanje brojaca i ukupnog prometa
    ++broj_musterija;
    ukupan_promet += ukupno;
}

```

```

    // Ispitivanje da li se nastavlja unos sledece musterije
    cout << endl<< endl;
    cout << "Da li ima jos neko ko hoce da plati ?" << endl;
    cout << "Ako ima kucaj 1 a ako nema kucaj 0 : ";
    cin >> odgovor;
}

// Ispisivanje ukupnog broja musterija i ukpnog prometa
cout << endl;
cout << "Ukpno : " << endl << endl;
cout << "Broj musterija : " << broj_musterija << endl;
cout << "Promet : " << ukupan_promet << endl;

// Ispisivanje završne poruke
cout << endl;
cout << "*** Zivi bili i dosli nam opet ! ***" << endl;
cout << endl;

return 0;
}

```

Zadatak 8. Napisati program koji izračunava mesečnu kamatu na štednju primenom **for** petlje. Uslov za izvršavanje petlje se nalazi iza ključne reči **for**. Primetimo da se ispisivanje rezultata i izračunavanja takođe vrše u okviru **for** petlje, što se često koristi u cilju racionalizacije programskog koda.

```

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

int main() {

    //Deklarisanje promenljivih
    double godisnja_kamata, // Godisnja kamata u %
           mesecna_kamata, // Mesecna kamata
           ulog,           // Visina uloga
           stanje_racuna, // Mesecno stanje racuna
           kamatni_iznos, // Kamatni iznos
           ukupna_kamata; // Ukupna kamata

    int mesec, // Racuna period prikazan na izvodu sa racuna
        period; // Period na izvodu sa racuna u mesecima

    //Podesavanje izlaznog formata za ispisivanja iznosa
    cout << setprecision(2)
          << setiosflags(ios::fixed)
          << setiosflags(ios::showpoint);

    // Unos podataka
    cout << "Unesite visinu uloga : ";
    cin >> ulog;
    cout << endl;

    cout << "Unesite period izrazen u broju meseci : ";
    cin >> period;
    cout << endl;
}

```

```

cout << "Unesite godisnju kamatu u % : ";
cin >> godisnja_kamata;

// Izracunavanja i ispisivanje rezultata
mesečna_kamata = (godisnja_kamata / 100) / 12;
ukupna_kamata = 0.00;
stanje_racuna = ulog;

cout << endl << endl;
cout << "           MESECNI           UKUPAN           NOVO" << endl;
cout << " MESEC      KAMATNI IZNOS      KAMATNI IZNOS      STANJE" << endl;
cout << "-----";

for (meseć = 1; meseć <= period; ++meseć) {
    kamatni_iznos = mesećna_kamata * stanje_racuna;
    stanje_racuna += kamatni_iznos;
    ukupna_kamata += kamatni_iznos;

    cout << endl << setw(4) << meseć
        << setw(14) << kamatni_iznos
        << setw(16) << ukupna_kamata
        << setw(14) << stanje_racuna;
}

cout << endl;
cout << "-----" << endl << endl;
cout << "\tUkupno" << endl << endl;
cout << "Početni ulog : " << setw(8) << ulog << endl;
cout << "Kamata : " << setw(8) << ukupna_kamata << endl;
cout << "Krajnja suma : " << setw(8) << stanje_racuna << endl << endl;

return 0;
}

```

Zadatak 9. Napisati program koji prebrojava znakove unete sa konzole, u više linija korišćenjem ugneždene **while** petlje. Ugneždene petlja se nalazi u okviru spoljašnje petlje i može se izvršiti više puta ili ni jednom ukoliko uslov ugneždene petlje nije ispunjen, u toku jednog izvršavanja spoljašnje petlje.

```

#include <iostream>
using namespace std;

int main() {

    // Deklarisanje promenljivih
    int znak; // Promenljiva u koju se smesta znak unet
              //od strane korisnika
    int broj_znakova = 0; // Broji unete karaktere

    // Ispisivanje poruke sa uputstvom za korisnika
    cout << "Ovaj program broji unete znakove sa tastaure." << endl
        << "Unesite neki broj znakova u svaki red" << endl
        << "a dobićete ukupan broj unetih znakova. " << endl
        << "Završite svaki red sa Enter." << endl

        << "Završite unos priskom na Enter, [Ctrl]+Z, a zatim Enter."

```



```

        << endl << endl;

// Unos i brojanje znakova
while ((znak = cin.get()) != EOF){ // Petlja koja se izvrsava
                                   //sve dok nije kraj unosa
    ++broj_znakova;
    while ((znak = cin.get()) != '\n') // Petlja koja se izvrsava
                                           // sve dok nije kraj reda
        ++broj_znakova;
}

// Ispisivanje rezultata
cout << endl;
cout << "Uneli ste " << broj_znakova << " znakova." << endl << endl;

return 0;
}

```

Zadatak 10. Napisati program koji crta pravougli trougao znacima 'X' korišćenjem ugneždene for petlje.

```

#include <iostream>
using namespace std;

int main() {

    // Deklarisanje promenljivih
    const int BROJ_REDOVA = 22;
    int trenutni_broj_koraka, brojac_x;

    // Ispisivanje rezultata
    for (trenutni_broj_koraka = 1; trenutni_broj_koraka <= BROJ_REDOVA;
        ++trenutni_broj_koraka) {
        cout << endl;
        for (brojac_x = 1; brojac_x <= trenutni_broj_koraka; ++brojac_x)
            cout << "X";
    }

    cout << endl << endl;

    return 0;
}

```

Zadatak 11. Napisati program koji računa platu za zaposlene. Uzeti u obzir da se prekovremeni rad plaća 50% više od redovnog. Predvideti mogućnost obrade podataka za više zaposlenih, kao i sumiranje plata na kraju programa.

```

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

int main() {

    // Deklarisanje promenljivih
    const double FAKTOR_PREKOVREMENOG_RADA = 1.5;
    const double REGULARAN_BROJ_SATI = 40.0;
    int brojac_zaposlenih,

```

```

        sledeci_zaposleni;        // 1 ako postoji sledeci;
                                   // 0 ako ne postoji
double radni_sati,
        satnica,
        regularna_plata,
        prekovremena_plata,
        ukupna_plata,
        sve_plate;

// Inicijalizacija promenljivih
sve_plate = 0.00;
brojac_zaposlenih = 0;

//Podesavanje izlaznog formata za ispisivanja iznosa
cout << setprecision(2)
        << setiosflags(ios::fixed)
        << setiosflags(ios::showpoint);

do { // Pocetak while petlje

    // Unos podataka
    cout << endl;
    cout << "Unesite broj ostvarenih radnih sati : ";
    cin >> radni_sati;

    cout << "\nUnesite satnicu : ";
    cin >> satnica;

    // Izracunavanje plate
    if (radni_sati > REGULARAN_BROJ_SATI) {
        regularna_plata = REGULARAN_BROJ_SATI * satnica;
        prekovremena_plata = (radni_sati - REGULARAN_BROJ_SATI)
            *FAKTOR_PREKOVREMENOG_RADA * satnica;
    }
    else {
        regularna_plata = radni_sati * satnica;
        prekovremena_plata = 0.00;
    }

    ukupna_plata = regularna_plata + prekovremena_plata;
    sve_plate += ukupna_plata;
    ++brojac_zaposlenih;

    // Ispisivanje plate
    cout << endl << endl;
    cout << "REGULARNO    PREKOVREMENO    UKUPNO";
    cout << endl << setw(9) << regularna_plata
        << setw(16) << prekovremena_plata
        << setw(11) << ukupna_plata << endl;
    cout << "-----" << endl;

    // Pitanje korisniku da li zeli da nastavi sa sledecim zaposlenim
    cout << endl << endl;
    cout << "Da li zelita da nastavite sa sledecim zaposlenim ?" << endl;
    cout << "Unesite 1 za Da ili 0 za Ne : ";
    cin >> sledeci_zaposleni;

```

```

}

while (sledeci_zaposleni); // Uslov while petlje - na kraju bloka naredbi
// Ispisivanje zbira svih plata
cout << endl << endl;
cout << "Ukupna plata za " << brojac_zaposlenih
    << " zaposlenih je " << sve_plate << endl << endl;

return 0;
}

```

Zadatak 12. Napisati program koji odlučuje da li je pritisnuto dugme cifra.

```

#include <iostream>
using namespace std;

int main() {

    // Deklarisanje promenljivih
    char znak;

    // Unos znaka sa tastature
    cout << "Pritisnite neko dugme na tastaturi, a zatim pritisnite Enter : ";
    znak = cin.get();
    cout << endl;

    // Provera unetog znaka i ispisivanje rezultata
    if ( (znak >= '0') && (znak <= '9'))
        cout << "Pritisnuli ste dugme koje je cifra." << endl;
    else
        cout << "Pritisnuli ste dugme koje nije cifra." << endl;

    return 0;
}

```

Zadatak 13. Napisati program koji izračunava plate zaposlenih u manjoj kompaniji. U programu moraju biti posebno obrađivani domaći, a posebno strani radnici. Potrebno je izračunati i porez na platu svakog zaposlenog, a na kraju prikazati zbirne podatke.

```

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

int main() {

    // Deklarisanje promenljivih
    const double KOEFICIJENT_ZA_DOMACE_RADNIKE = 0.070;
    const double KOEFICIJENT_ZA_STRANE_RADNIKE = 0.045;
    int broj_zaposlenih;
    char kod_zaposlenog,
        kod_drzave,
        odgovor;
    double satnica,
        radni_sati,
        regularni_rad,

```

```

        prekovremeni_rad,
        ukupna_plata,
        porez,
        neto_plata,
        suma_ukupno_plate,
        suma_porez,
        suma_neto_plata;

//Podesavanje izlaznog formata za ispisivanja iznosa
cout << setprecision(2)
    << setiosflags(ios::fixed)
    << setiosflags(ios::showpoint);

// Ispisivanje poruke
cout << "Ovaj program racuna neto platu za svakog zaposlenog." << endl;
cout << "Pri svakom pitanju, unesite trazene podatke.";

// Inicijalizacija zbirova
broj_zaposlenih = 0;
suma_ukupno_plate = suma_porez = suma_neto_plata = 0.00;

// Glavna petlja
do { // Pocetak while petlje
    // Unos podataka
    cout << endl << endl << "Unesite satnicu : ";
    cin >> satnica;

    cout << endl << "Unesite broj ostvarenih radnih sati : ";
    cin >> radni_sati;

    cout << endl << "Zaposleni moze biti oslobodjen placanja poreza." << endl;
    cout << "Ako je oslobodjen placanja poreza unesite 'O' a ako "
        << "nije unesite 'N' : ";
    cin.get();

    kod_zaposlenog = cin.get();

    cout << endl << "Zaposleni moze biti domaci ili strani drzavljanin." << endl;
    cout << "Ako je domaci drzavljanin unesite 'D' a ako je "
        << "strani drzavljanin unesite 'S' : ";
    cin.get();

    kod_drzave = cin.get();

    // Racunanje ukupne plate
    if (radni_sati > 40.0) {
        regularni_rad = 40.0 * satnica;
        prekovremeni_rad = (radni_sati - 40.0) * 1.5 * satnica;
    }
    else {
        regularni_rad = radni_sati * satnica;
        prekovremeni_rad = 0.00;
    }

    ukupna_plata = regularni_rad + prekovremeni_rad;

```

```

// Racunanje poreza
if ((kod_zaposlenog == 'N') || (kod_zaposlenog == 'n'))
    if ((kod_drzave == 'D') || (kod_drzave == 'd'))
        porez = ukupna_plata * KOEFICIJENT_ZA_DOMACE_RADNIKE;
    else
        porez = ukupna_plata * KOEFICIJENT_ZA_STRANE_RADNIKE;
else
    porez = 0.00;

// Racunanje neto plate
neto_plata = ukupna_plata - porez;

// Ispisivanje rezultata
cout << endl << endl;
cout << "Redovna plata :" << setw(10) << regularni_rad << endl;
cout << "Prekovremeno : " << setw(10) << prekovremeni_rad << endl;
cout << "Porez :          " << setw(10) << porez << endl;
cout << "Ukupno :          " << setw(10) << ukupna_plata <<
    endl;
cout << "-----" << endl;
cout << "Neto plata :      " << setw(10) << neto_plata << endl;

// Povecanje zbroja
++broj_zaposlenih;
suma_ukupno_plate += ukupna_plata;
suma_porez += porez;
suma_netto_plata += neto_plata;

// Odluka o unosu sledeceg zaposlenog
cout << endl << endl;
cout << "Zelite li da nastavite sa sledecim zaposlenim ? ( d/n): ";
cin.get();

    odgovor = cin.get();
} while ( (odgovor == 'd') || (odgovor == 'D') ); // Uslov while petlje

// Ispisivanje suma
cout << endl << endl << endl;
cout << "Ukupan broj zaposlenih : " << setw(8) << broj_zaposlenih << endl;
cout << "Zbir ukupnih plata : " << setw(12) << suma_ukupno_plate << endl;
cout << "Zbir poreza :          " << setw(12) << suma_porez << endl;
cout << "Zbir neto plata :      " << setw(12) << suma_netto_plata << endl;

return 0;
}

```

Zadatak 14. Napisati program koji izračunava komisionu cenu u zavisnosti od tipa robe.

```

#include <iostream>
#include <iomanip>
#include <cstdlib>
using namespace std;

int main() {
    // Deklarisanje promenljivih
    const double    DOMACA_STOPA = 0.060;

```

```

const double    REGIONALNA_STOPA = 0.050;
const double    KOMERCIJALNA_STOPA = 0.045;
int            kod_robe;
double         prodajna_cena,
              komisiona_stopa,
              komisiona_cena;

//Podesavanje izlaznog formata za ispisivanja iznosa
cout << setprecision(2)
      << setiosflags(ios::fixed)
      << setiosflags(ios::showpoint);

// Unos podataka
cout << "Unesite prodajnu cenu proizvoda: ";
cin >> prodajna_cena;

cout << endl;
cout << "Unesite sifru proizvoda na sledeci nacin." << endl << endl;
cout << "Domaci proizvod,          unesite D" << endl;
cout << "Regionani proizvod,        unesite R" << endl;
cout << "Komerrijalni proizvod, unesite C" << endl << endl;
cout << "Izaberite opciju : ";
cin.get();

kod_robe = cin.get();

// Izracunavanja
switch (kod_robe) {

    case 'D':
    case 'd':
        komisiona_stopa = DOMACA_STOPA;
        break;

    case 'R':
    case 'r':
        komisiona_stopa = REGIONALNA_STOPA;
        break;

    case 'C':
    case 'c':
        komisiona_stopa = KOMERCIJALNA_STOPA;
        break;

    default:
        cout << endl << endl << "Neispravna sifra proizvoda! Pokusajte ponovo"
              << endl;
        exit(1);
        break;
}

komisiona_cena = prodajna_cena * komisiona_stopa;

// Ispisivanje rezultata
cout << endl << endl;
cout << "Komisiona cena je " << komisiona_cena << endl << endl;

```

```
    return 0;  
}
```

Funkcije

Zadatak 15. Napisati program koji poziva funkciju definisanu van funkcije **main()** i iscrtava pravougaonik znacima 'X' pri čemu koristi funkciju definisanu izvan funkcije **main()**.

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

void Napisi_10_puta_po_X();    // Deklarisanje funkcije

int main() {

    // Deklarisanje promenljivih
    int i;

    // Pozivanje funkcije za ispisivanje kroz petlju
    for (i = 1; i <= 5; ++i) {
        Napisi_10_puta_po_X();
        cout << endl;
    }

    cout << endl;

    return 0;
}

void Napisi_10_puta_po_X()    // Definisiranje funkcije
{
    // Deklarisanje promenljivih
    int j;

    // Ispisivanje 10 znakova 'X' u jednom redu kroz petlju
    for (j = 1; j <= 10; ++j)
        cout << 'X';
}
```

Zadatak 16. Napisati program koji poziva funkciju definisanu van funkcije **main()** i crta trougao izabranim znacima.

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

void Napisi_Znak(char, int);    // Deklarisanje funkcije

int main() {

    // Deklarisanje promenljivih
    int i, broj_redova;
    char znak;

    // Unos podataka
    cout << "Unesite znak koji zelite da bude prikazan : ";
    cin >> znak;
    cout << endl;
```



```

cout << "Unesite broj redova koji zelite da bude prikazan : ";
cin >> broj_redova;

// Pozivanje funkcije za ispisivanje kroz petlju
cout << endl;

for (i = 1; i <= broj_redova; ++i) {
    Napisi_Znak(znak, i);
    cout << endl;
}

cout << endl;

return 0;
}

void Napisi_Znak(char izabrani_znak, int brojac)    // Definisanje funkcije
{
    // Deklarisanje promenljivih
    int j;

    // Ispisivanje izabranog znaka izabrani broj puta u jednom redu kroz petlju
    for (j = 1; j <= brojac; ++j)
        cout << izabrani_znak;

    return;
}

```

Zadatak 17. Napisati program koji prosledjuje argument funkcije po vrednosti i analizirati povratnu vrednost funkcije.

```

#include <iostream>
using namespace std;

void Funkcija(int);    // Deklarisanje funkcije

int main() {

    // Deklarisanje promenljivih
    int i;
    // Inicijalizacija promenljive i na 5
    i = 5;

    // Ispisivanje vrednosti promenljive i pre poziva funkcije
    cout << "Pre poziva fukcije, i = " << i << endl;

    // Vrednost i se prosledjuje funkciji Funkcija(),
    // i dodeljuje parametru argument (videti u definiciji funkcije
    // Funkcija()).
    // Funkcija povecava vrednost svoje promenljive argument za 1
    // i vraca izvrsavanje programa nazad u main().
    // Vrednost promenljive i se ne menja pozivom funkcije Funkcija().

    Funkcija(i);
}

```

```

    // Ispisivanje vrednosti promenljive i posle poziva funkcije
    cout << "Posle poziva funkcije, i = " << i << endl;
    cout << endl;

    return 0;
}

void Funkcija(int argument)    // Definisavanje funkcije
{
    // Vrednost prosledjena funkciji se dodeljuje
    // parametru argument. Funkcija povecava vrednost
    // promenljive argument za jedan i vraca kontrolu funkciji main().
    ++argument;

    return;
}

```

Zadatak 18. Napisati program koji izračunava ocenu na ispitu na osnovu broja bodova sa usmenog i pismenog dela ispita. Koristiti funkciju definisanu izvan funkcije main().

```

#include <iostream>
using namespace std;

int Izracunaj_Konacnan_Broj_Bodova(int, int);    // Deklarisanje funkcije

int main() {

    // Deklarisanje promenljivih
    int bodovi_sa_pismenog_ispita,
        bodovi_sa_usmenog_ispita,
        konacan_broj_bodova;

    // Unos podataka
    cout << "Unesite broj bodova sa pismenog ispita : ";
    cin >> bodovi_sa_pismenog_ispita;
    cout << endl;

    cout << "Unesite broj bodova sa usmenog ispita : ";
    cin >> bodovi_sa_usmenog_ispita;

    // izracunavanja pozivom funkcije
    konacan_broj_bodova = Izracunaj_Konacnan_Broj_Bodova(bodovi_sa_pismenog_ispita,
        bodovi_sa_usmenog_ispita);

    // Ispisivanje rezultata
    cout << endl;
    cout << "Konacan broj bodova je " << konacan_broj_bodova << endl;
    cout << endl;

    return 0;
}

int Izracunaj_Konacnan_Broj_Bodova(int pismeni, int usmeni) // Definisavanje funkcije
{
    // Deklarisanje promenljivih
    const double UTICAJ_PISMENOG_ISPITA = 0.40;

```

```

const double UTICAJ_USMENOG_ISPITA = 0.60;
double bodovi;
int zaokruženi_bodovi;

// Izracunavanja
bodovi = UTICAJ_PISMENOG_ISPITA * pismeni + UTICAJ_USMENOG_ISPITA * usmeni;
zaokruženi_bodovi = bodovi + 0.5;

return zaokruženi_bodovi;
}

```

Zadatak 19. Napisati program koji demonstrira upotrebu lokalnih i globalnih promenljivih.

```

#include <iostream>
using namespace std;

// Deklarisanje globalnih promenljivih
int globalna_promenljiva_1,
    globalna_promenljiva_2;

void Funkcija(); // Deklarisanje funkcije

int main() {
    // Sledeca dodeljivanja vrednosti su legalna jer su promenljive
    // globalna_promenljiva_1 i globalna_promenljiva_2
    // globalne promenljive deklarirane pre funkcije main().
    globalna_promenljiva_1 = 88;
    globalna_promenljiva_2 = 99;

    // Ispisivanje vrednosti globalnih promenljivih
    // globalna_promenljiva_1 i globalna_promenljiva_2.
    cout << "Inicijalne vrednosti globalnih promenljivih :" << endl;
    cout << "globalna promenljiva 1 = " << globalna_promenljiva_1
        << " globalna promenljiva 2 = " << globalna_promenljiva_2
        << endl << endl;

    // Poziv funkcije Funkcija()
    Funkcija();
    // Ispisivanje vrednosti globalnih promenljivih
    // globalna_promenljiva_1 i globalna_promenljiva_2
    // nakon poziva funkcije Funkcija().
    cout << endl;
    cout << "Nakon poziva funkcije Funkcija(), globalne promenljive su :" << endl;
    cout << "globalna promenljiva 1 = " << globalna_promenljiva_1
        << " globalna promenljiva 2 = " << globalna_promenljiva_2
        << endl << endl;

    return 0;
}

// Deklarisanje globalne promenljive
int globalna_promenljiva_3;

void Funkcija() // Definisane funkcije
{
    //Deklarisanje lokalnih promenljivih

```

```

int globalna_promenljiva_1,    // Lokalna promenljiva za funkciju Funkcija()
    lokalna_promenljiva_1;    // Lokalna promenljiva

// globalna_promenljiva_1 je razlicita od globalne promenljive
// globalna_promenljiva_1 koja je deklarisan pre funkcije main().

globalna_promenljiva_3 = 1; // Dodeljivanje vrednosti globalnoj promenljivoj 3

lokalna_promenljiva_1 = 2; // Inicijalizacija lokalne promenljive
    //lokalna_promenljiva_1

globalna_promenljiva_1 = 3; // Ne menja vrednost promenljive
    // globalna_promenljiva_1 u main() funkciji.

globalna_promenljiva_2 = 4; // Posto globalna_promenljiva_2 nije deklarisan
//u ovoj funkciji, radi se o globalnoj promenljivoj deklarisanom pre
// main() funkcije. Zbog toga, ova dodela vrednosti menja vrednost
// promenljive globalna_promenljiva_2 u main() funkciji.
// Ispisivanje vrednosti lokalnih promenljivih
// globalna_promenljiva_1 i lokalna_promenljiva_1
// i globalnih promenljivih globalna_promenljiva_2 i globalna_promenljiva_3.

cout << "U funkciji Funkcija(), globalne promenljive su :"<< endl;
cout << "globalna promenljiva 2 = " << globalna_promenljiva_2
    << " globalna promenljiva 3 = " << globalna_promenljiva_3
    << endl << endl;

cout << "U funkciji Funkcija(), lokalne promenljive su : " << endl;
cout << "globalna promenljiva 1 = " << globalna_promenljiva_1
    << " lokalna promenljiva 1 = " << lokalna_promenljiva_1 << endl;
}

```

Zadatak 20. Napisati program koji broji iteracije u petlji primenom **static** promenljive.

```

#include <iostream>
using namespace std;

void Brojac_Iteracija(); // Deklarisanje funkcije

int main() {

    // Deklarisanje promenljivih
    int i;

    // for petlja koja poziva funkciju Brojac_Iteracija()
    // definisani broj puta
    for (i = 1; i <= 5; ++i)
        Brojac_Iteracija();

    cout << endl;

    return 0;
}

```

```

void Brojac_Iteracija()    // Definicija funkcije
{
    // Deklarisanje promenljivih
    static int brojac = 0;    // static oznacava da je memorijska
    // lokacija u koju se smesta promenljiva brojac nepromenljiva

    ++brojac;

    // Ispisivanje poruke
    cout << "Iteracija broj " << brojac << endl;

    return;
}

```

Zadatak 21. Napisati program koji izračunava cenu selidbe nameštaja u zavisnosti od težine i udaljenosti na koju se prevozi. U programu koristiti funkcije, kao i globalne i lokalne promenljive.

```

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

double Izracunavanje_Cene_Rada(int);

double Izracunavanje_Cene_Prevoza(int);

void Ispisivanje_Troskova(double, double, double);

int main() {

    // Deklarisanje promenljivih
    int    tezina,          // Tezina nameštaja
           rastojanje;     // Rastojanje za prevoz

    double troskovi_rada,   // Troskovi prenosjenja nameštaja
           troskovi_prevoza, // Troskovi prevoza
           ukupni_troskovi; // Ukupni troskovi

    //Podesavanje izlaznog formata za ispisivanja iznosa
    cout << setprecision(2)
          << setiosflags(ios::fixed)
          << setiosflags(ios::showpoint);

    // Unos podataka
    cout << "Unesite tezinu u kilogramima : ";
    cin >> tezina;
    cout << endl;

    cout << "Unesite rastojanje u kilometrima : ";
    cin >> rastojanje;

    // Izracunavanja pozivom funkcija
    troskovi_rada = Izracunavanje_Cene_Rada(tezina);
    troskovi_prevoza = Izracunavanje_Cene_Prevoza(rastojanje);
    ukupni_troskovi = troskovi_rada + troskovi_prevoza;

    // Ispisivanje rezultata pozivom funkcije

```

```

    Ispisivanje_Troskova(troskovi_rada, troskovi_prevoza, ukupni_troskovi);
    cout << endl;

    return 0;
} // Kraj funkcije main()

double Izracunavanje_Cene_Rada(int tezina) {

    // Deklarisanje promenljivih
    const double KOEFICIJENT_RADA = 4.00;
    double troskovi_rada;

    // Izracunavanja
    troskovi_rada = (tezina / 100) * KOEFICIJENT_RADA;

    return troskovi_rada;
} // Kraj funkcije Izracunavanje_Cene_Rada()

double Izracunavanje_Cene_Prevoza(int rastojanje) {

    // Deklarisanje promenljivih
    const double KOEFICIJENT_PREVOZA = 1.75;
    const double FIKSNI_TROSKOVI_PREVOZA = 50.00;
    double troskovi_prevoza;

    // Izracunavanja
    troskovi_prevoza = FIKSNI_TROSKOVI_PREVOZA + rastojanje * KOEFICIJENT_PREVOZA;

    return troskovi_prevoza;
} // Kraj funkcije Izracunavanje_Cene_Prevoza()

void Ispisivanje_Troskova(double cena_rada, double cena_prevoza,
    double ukupni_troskovi) {

    // Ispisivanje rezultata
    cout << endl;
    cout << "Cena selidbe je :" << endl << endl;
    cout << "Prenosenje      : " << setw(9) << cena_rada << endl;
    cout << "Prevoz            : " << setw(9) << cena_prevoza << endl;
    cout << "-----" << endl;
    cout << "Ukupno              : " << setw(9) << ukupni_troskovi;
}

```

Zadatak 22. Napisati program koji izračunava mesečnu ratu prilikom otplate kredita upotrebom funkcija.

```

#include <iostream>
#include <iomanip>
#include <cstdlib>

using namespace std;

void Ispisivanje_Pozdravne_Poruke();

double Unos_Visine_Kredita();

double Unos_Kamatne_Stope();

```



```

int Unos_Roka_Otplate() {

    const int MAX_ROK = 5;
    int rok;

    cout << endl;
    cout <<"Unesite broj godina otplate : ";
    cin >> rok;

    if (rok > MAX_ROK) {
        cout << endl;
        cout << "Zao nam je, rok otplate je veci od"
            << MAX_ROK << "godina" << endl;
        exit(1);
    }
    else

        return rok;

}

```

Zadatak 23. Napisati program koji stepenuje uneti broj željenim eksponentom.

```

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

int main() {

    double osnova;
    double eksponent;

    cout << "Unesite osnovu : ";
    cin >> osnova;

    cout << "Unesite eksponent : ";
    cin >> eksponent;

    cout << endl << osnova << " na stepen " << eksponent
        << " je " << pow(osnova, eksponent) << endl << endl;

    return 0;
}

```

Zadatak 24. Napisati program koji pronalazi kvadratni koren unetog broja.

```

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

int main() {

    double broj;

```

```
cout << "Unesite broj ciji kvadratni koren zelite da nadjete : ";
cin >> broj;

cout << endl << "Kvadratni koren broja " << broj
    << " je " << sqrt(broj) << endl << endl;

return 0;
}
```