**IZGLED C++ PROGRAMA**

****

**Kod kuće**

<https://sourceforge.net/projects/orwelldevcpp/>

**1. Otvoriti DEV C++**

**2. File – NEW – SOURCE FILE (ne project!!!)**

**3. Nakucati program**

**4. Pokrenuti program tipkom F11**

**5. Spremiti program pod nekim nazivom**

#include <iostream>

#include <cstdlib>

using namespace std;

int main ( )

{

 cout << “ PRVI PROGRAM!!! “ << endl;

system (''PAUSE'');

return 0;

}

**#include <iostream>**

#include – najava biblioteka iz kojih ćemo koristiti naredbe

iostream – biblioteka naredbi koje omogućuju vezu programa s tipkovnicom i monitorom

**#include <cstdlib>**

- biblioteka standardnih naredbi

**using namespace std;** aktivira imenik

**int** **main( ) - glavna funkcija**

{ početak bloka naredbi

} kraj bloka naredbi

; kraj svake naredbe

**system (''PAUSE''**) – pauziranje programa (bez gašenja) dok se ne pritisne neka druga tipka

**return 0** – povratna vrijednost glavnoj funkciji

**endl ili \n prelazak u novi red**

**cout << naredba za ispis podataka iz programa na ekran**

**cin >> naredba za unos podataka s tipkovnice**

**\t razmak, tab**

**//** komentar u jednom redu

**/\*** blok komentara **\*/ u više redova**

VARIJABLA – nepoznanica u programu:

**int = cijeli broj**

**float = decimalni broj**

**char = jedan znak (a, A, $...)**

**string = niz znakova (riječ ili rečenica…)**

C++ ima svoju glavnu funkciju (glavni dio programa) pod simboličkim imenom **main( ).**

Izvođenje svakog C++ programa zapravo počinje naredbama koje se nalaze u njoj.

int = cijeli broj (integer) [kao rezultat te funkcije vratit će se cijeli broj]

 main ( ) = u zagradu pišemo parametre (argumente) funkcije (to su podaci koji se prenose, odnosno unose u funkciju main)

{ početak bloka naredbi

} kraj bloka naredbi

; taj znak mora zaključivati svaku naredbu

Na početku programa moramo **deklarirati biblioteke** (standardne već gotove procedure, tako zvane datoteke zaglavlja), koje nam omogućuju rad s tipkovnicom, zaslonom monitora, standardnim naredbama...

**#include <iostream>**

#include – pretprocesorska naredba (sve započinju znakom #), najava biblioteka iz kojih ćemo koristiti naredbe

iostream – u toj biblioteci smještene su naredbe koje omogućuju vezu programa s tipkovnicom i monitorom.

**#include <cstdlib>**

- biblioteka standardnih naredbi

(npr, za matematičke složenije operacije koristimo biblioteku #include <math.h> ili <cmath>)

**using namespace std;** ta naredba aktivira imenik (namespace = imenik) pod imenom std, koji sadrži sve standardne funkcije (uvedena je da se izbjegnu ista imena naredbi, funkcija u različitim bibliotekama)

* omogućuje nam korištenje svih naredbi u prostoru .std

**system (''PAUSE''**) – pauziranje programa (bez gašenja) dok se ne pritisne neka druga tipka

**return 0** – povratna vrijednost glavnoj funkciji - funkcija se prestaje izvršavati dolaskom na naredbu return

**endl ili \n prelazak u novi red**

**cout naredba za ispis podataka iz programa na ekran**

**cin naredba za unos podataka s tipkovnice**

**\t razmak, tab**

**//** komentar u jednom redu

**/\*** blok komentara **\*/ u više redova**

- komentari služe kao uputa ili podsjetnik programera da zna što je napravio; komentari se ne kompajliraju i ne pretvaraju u program

**ISPIS**

 **cout << ''tekst... '';**

**cout** – izlazna naredba vezana uz zaslon računala (c=jezik c, out = output)

**<<** - podatak koji slijedi upućuje se na zaslon

**'' ''** - unutar navodnika nalazi se tekst koji treba ispisati doslovce

**Da bi prešli u novi red** koristimo naredbu:

 **endl** odnosno **cout <<endl;**

primjer ispisa nekog teksta i nakon toga skok u novi red:

 cout <<''tekst... '' << endl;

 ili

 cout <<''tekst... '';

 cout << endl;

Ako želimo više redaka razmaka, više puta napišemo tu naredbu!

primjer:

 cout <<''tekst ... '' <<endl <<endl <<endl <<endl;

 ili

 cout <<endl;

 cout <<endl;

 cout <<endl;

 cout <<endl;

 ...

**UNOS**

**cin >> ime varijable;**

**cin** – učitavanje podataka s tipkovnice (in=input)

**>>** - ovim operatorom podaci s tipkovnice upućuju se varijabli

**ime varijable** – određujemo sami u skladu s pravilima o imenu

**NAPOMENA:** varijable koje ćemo koristiti moramo najaviti (deklarirati) na početku glavnog dijela programa (tip i ime) – time smo rezervirali određeni memorijski prostor u koji ćemo pohranjivati vrijednosti tih varijabli!

**VAŽNO: moramo razlikovati << od >> (operator ispisa od operatora unosa)!**

primjer: cin >> a;

**X**

**int = cijeli broj**

**float = decimalni broj**

**char = jedan znak (a,$...)**

**string = niz znakova (riječ ili rečenica…)**

**#include <iostream>**

**#include <cstdlib>**

**using namespace std;**

**int main ( )**

**{**

 **int x,y; // UPOZNAVANJE SA NEPOZNANICAMA, deklaracija varijabli**

 **cout <<"Unesi dva broja, a ja ću ih ispisati:"<<endl;**

 **cin >> x >> y;**

 **cout << "Unijeli ste broj: " << x <<"i" <<y << endl;**

 **system ("PAUSE");**

**return 0;**

**}**

1. *Napisati program koji zbraja dva broja koja korisnik unese i ispisuje rezultat.*
2. *Napisati program koji nakon što upišete godinu rođenja, računa i ispisuje koliko imate godina.*
3. *Napisati program koji nakon što korisnik unese jedan broj, izračuna i ispiše prethodnik tog broja i sljedbenik tog broja.*

**PRIDRUŽIVANJE VRIJEDNOSTI**

Varijabli možemo pridružiti vrijednost na dva načina:

1. **izvan programa** naredbom za **unos**

 cin >> ime varijable;

1. **unutar programa** naredbom za **dodjeljivanje** **(=)**

 ime varijable **=** vrijednost (izraz);

 primjer: int A=5;

 D=B+C;

 Prvi primjer čita se: vrijednost 5 smjestili smo u varijablu

 pod imenom A.

 Drugi primjer: prvo izračunamo vrijednost izraza na desnoj

 strani, a zatim dobiveni rezultat smjestimo u

 varijablu na lijevoj strani znaka =.

**Varijable koristimo za spremanje vrijednosti podataka. Računalo pamti samo zadnji sadržaj varijable. Vrijednost varijable može se mjenjati.**

**ARITMETIČKE OPERACIJE**

Imamo dva osnovna tipa numeričkih podataka:

1. **cjelobrojni**
2. **realni**

Osnovne računske operacije koje se izvode nad oba tipa podataka (tu se koriste **binarni operatori +, -, \*, /** [binarni znači da su potrebna dva objekta]):

* 1. **zbrajanje (+)**
	2. **oduzimanje (-)**
	3. **množenje (\*)**
	4. **dijeljenje (/)**

Nad cjelobrojnim tipom podatka možemo izvesti i:

 **ostatak cjelobrojnog dijeljenja – modulo (%)**

Kod Dev-C++ koriste se i **unarni operatori** koji djeluju samo na jedan objekt. Osim unarnog plusa (+a) i unarnog minusa (-a) koji mijenjaju predznak broja, koristi se i operator ++a (uvećat će vrijednost varijable a za 1) i operator - - a (umanjit će vrijednost varijable a za 1).

Primjer varijanti unarnog operatora:

{

 int a = 5;

 cout << +a << endl; // rezultat je 5

 cout << -a << endl; // rezultat je -5

 cout << ++a << endl; // rezultat je 6

 cout << - - a << endl; // rezultat je 5 (6-1)

 cout << a++ << endl; // prvo ispisuje (5), a onda uvećava

 cout << a << endl; // sada je rezultat 6

 cout << a - - << endl; // prvo ispisuje (6), a onda umanjuje

 cout << a << endl; // sada je rezultat 5

 system(''PAUSE'');

 return 0;

}

**Inkrement je povećanje za 1 (++).** **Dekrement je umanjenje za 1 (- -).**

1. *Napisati program koji nakon što upišete godinu rođenja, računa i ispisuje koliko imate godina.*
2. *Napisati program koji nakon što korisnik unese jedan broj, izračuna i ispiše prethodnik tog broja i sljedbenik tog broja.*
3. *Izračunaj i ispiši opseg kružnice 2rPI*
4. *Napiši program koji će vam kada unesete koliko imate kuna, ispisati koliko je to u eurima.*

**ARITMETIČKE (BROJČANE) FUNKCIJE**

**Obavezno najaviti na početku programa predprocesorsku naredbu –** **biblioteka matematičkih funkcija cmath:**

**#include <cmath>**

Opći oblik funkcija:

 **naziv funkcije (argument)**

Argumenti su vrijednosti koje dajemo funkciji.

**Neke od funkcija:**

**abs( )** – apsolutna vrijednost abs(int x)

**asin( )** – sinus nekog kuta (rad) asin(float x) [argument u rasponu od -1 do 1]

**sqrt( )** – kvadratni korjen sqrt(float x)

**ceil( )** – zaokružuje argument na najbliži veći cijeli broj ceil(float x)

**floor( )** – zaokružuje argument na najbliži manji cijeli broj floor(float x)

primjer: float broj = 123.65;

 cout << ceil (broj) << endl; // rezultat je 124

 cout << floor (broj) << endl; // rezultat je 123

**modf( )** – rastavlja broj na njegov cijeli i decimalni dio modf(double x)

**pow( )** – potencija pow(float x)