#include<….>– dyrektywa, dzięki której mamy możliwość dołączania plików do własnego programu. Jeśli mamy potrzebę dołączenia więcej niż jednego pliku - kilkukrotnie wykorzystujemy wyżej wspomnianą dyrektywę, a za kropki wstawiamy nazwę dołączanego pliku. Każdą dyrektywę piszemy w nowym wierszu.

printf- służy do wypisywania tekstu na ekranie

int main() { } – główna część każdego programu. Nawiasy klamrowe zaczynają i kończą blok programu(w nich umieszczamy instrukcje wykonywane w funkcji).

using namespace std - informujemy kompilator, że chcemy aby wszystkie funkcje, klasy i szablony należące do przestrzeni nazw nie wymagały przedrostka std.

Komentarz krótki, czyli taki, który w całości znajduje się w jednym wierszu rozpoczyna się znakami //. Wszystko, co znajduje się za nimi jest komentarzem,

Komentarz długi - jeśli komentarz zajmuje więcej niż jeden wiersz, należy jego początek oznaczyć znakami /*, a koniec — znakami */

return 0 – instrukcja oznaczająca zakończenie – przerwanie funkcji.

cout – console out – instrukcja wyprowadzająca na konsolę napis umieszczony w cudzysłowie.

<< - operator wyprowadzania napisu – musi stać między cout a napisem.

endl – powoduje przejście do nowejlini(można użyć zamiast tego polecenia \n ale wewnątrz cudzysłowie zaraz po napisie).

; - średnik – kończy każdą instrukcję

Zmienna, jak sama nazwa wskazuje będzie się zmieniać w trakcie programu. Zmienna to pewien stosunkowo mały obszar w pamięci, w którym możemy przechowywać dane różnego typu np. liczby całkowite, liczby rzeczywiste (zmiennoprzecinkowe), znak, tekst oraz kilka innych wartości, które będą nas w przyszłości interesowały. Nie można jednak wszystkiego zapisywać do jednej zmiennej. Każda zmienna ma swoje przeznaczenie, wielkość i właściwości. Na zmiennych liczbowych możemy wykonywać operacje matematyczne, w innych z kolei możemy przechowywać tekst. Zmienne deklarujemy podając typ i potem nazwę zmiennej np. ***int suma.*** Nazwa

zmiennej może składać się wyłącznie z liter, cyfr i znaków podkreślenia. Pierwszym znakiem nazwy zmiennej musi być wielka lub mała litera. Nie można używać liter z ogonkami i akcentami. Nie można używać spacji.

if(warunek) – instrukcja warunkowa, która sprawdza warunek zapisany w nawiasie, który musi być spełniony aby wykonał się kod zamieszczony w klamrach. Jeżeli warunek nie zostanie spełniony kod między {} zostanie pominięty przez program.

else – zostanie wykonany kod, jeżeli poprzedni warunek (if) nie zostanie spełniony.

pętla **for – for(**i*nicjacja*(od czego zaczynamy) ; *warunek*; *inkrementacja*(co ma się dziać w każdej iteracji)**)-** instrukcja do powtarzania instrukcji x – razy. Pętli for używa się wtedy, gdy liczba powtórzeń (iteracji) jest znana. Pętla będzie wykonywać instrukcje zamknięte w klamrach dopóki warunek będzie prawdziwy. Jeśli warunek nie będzie spełniony nastąpi zerwanie pętli.

pętla **while** – while(warunek) {}- pętla sprawdza warunek. Wszystko, co znajduje się między klamrami będzie wykonywane dotąd, aż warunek zostanie spełniony.

pętla **do while -** ta pętla jest bardzo podobna do poprzedniej. Różni się od pętli while tylko tym, że jej warunek zostaje sprawdzony na końcu, a nie na początku. Dzięki temu instrukcje pętli na pewno zostaną wykonane przynajmniej raz.

do
{
}while(*warunek***)**