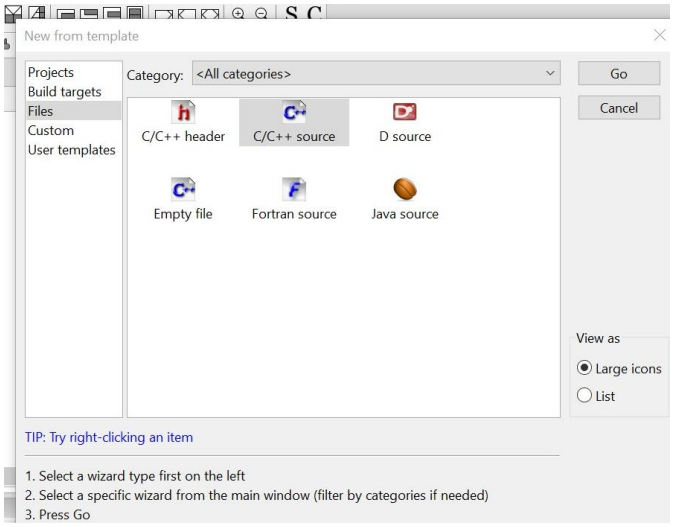
Laboratorium 1: Łagodne wprowadzenie do C

Czego się nauczymy:

Kompilacja i uruchomienie pierwszego programu w C (Code::Blocks)

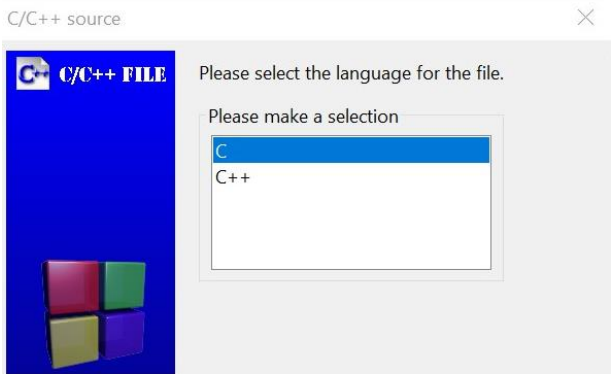
Zadanie 1 Proszę powtórzyć kroki opisane w poniższej instrukcji.

1. Otwieramy Code::Blocks
2. Tworzymy nowy plik: File->New->File
   a. Wybieramy C/C++source

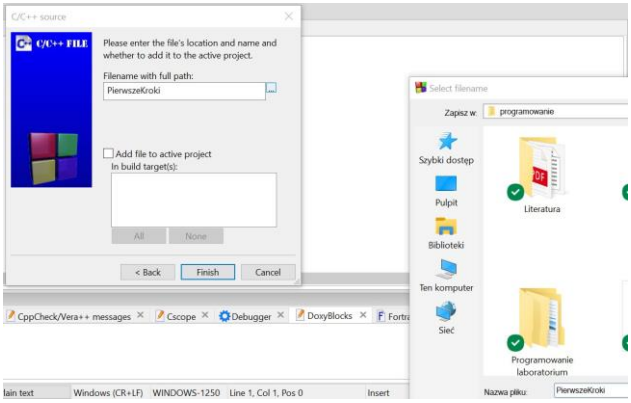


   b. Językiem, w którym napiszemy pierwszy program będzie C: u

c.



d. Nadajemy nazwę plikowi z kodem źródłowym i zapisujemy go w określonej lokalizacji

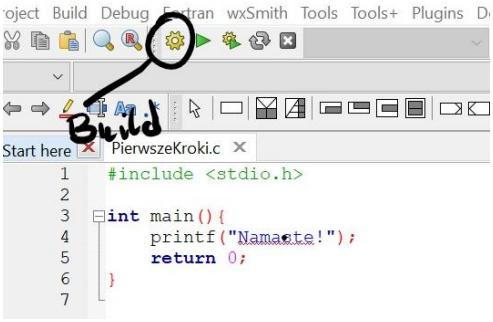


3. Piszemy pierwszy program, wyświetlający na ekranie prosty komunikat: „Namaste!”:

```c
#include <stdio.h>

int main(){
    printf("Namaste!");
    return 0;
}
```

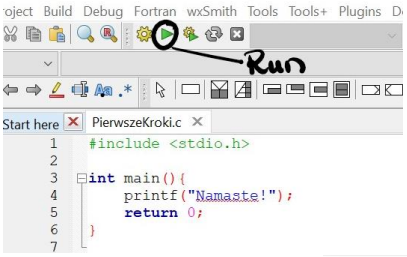
4. Kompilujemy

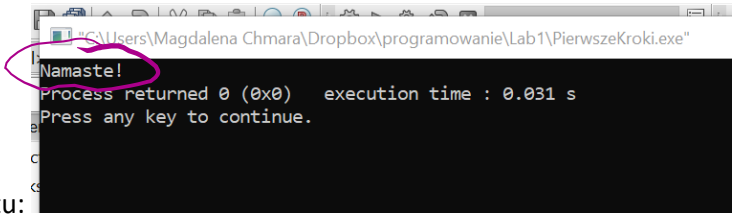


a. Sprawdzamy, że powstały dwa nowe pliki (.o i .exe), a w logu Build log wyświetlił się odpowiedni komunikat (jeśli log nie jest widoczny kliknij F2):

```
gcc.exe   -c "C:\Users\Magdalena Chmara\Dropbox\programowanie\Lab1\PierwszeKroki.c" -o "C:\Users\Magdalena Chmara\Dropbox\programowanie\Lab1\PierwszeKroki.o"
gcc.exe  -o "C:\Users\Magdalena Chmara\Dropbox\programowanie\Lab1\PierwszeKroki.exe" "C:\Users\Magdalena Chmara\Dropbox\programowanie\Lab1\PierwszeKroki.o"
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

5. Uruchamiamy





a. Cieszymy się z efektu:

b. Po zakończeniu zadania zamykamy czarne okienko naciskając dowolny klawisz.

```c
#include <stdio.h>

int main(){
    printf("Namaste!");
    return 0;
}
```

Budowa pierwszego programu (main, printf, return)

- Każda instrukcja w C kończy się średnikiem ;
- **main()** jest główną funkcją w każdym programie w C. Od niej kompilator rozpoczyna wykonywanie programy. Treść tej funkcji (ciało) zawarta jest między dwoma nawiasami klamrowymi: {}
- **#include** - dołącza do programu biblioteki zawierające różne dodatkowe funkcje. Tutaj dołączamy bibliotekę **stdio** („STandarD Input/Output" – standardowe wejście i wyjście (np. (wejście to np. wczytywanie informacji z klawiatury, a wyjście wyświetlanie tekstu na ekranie). Biblioteki umieszczone są w plikach z rozszerzeniem „.h" i zwane są inaczej plikami nagłówkowymi (h jak header-nagłówek). Proces przyłączenia funkcji bibliotecznych nosi nazwę linkowania.
- Jedną z funkcji pochodzących z biblioteki **stdio** funkcja **printf()**, która wyświetla na ekranie tekst zamieszczony w cudzysłowach.
- **return 0 –** jeśli wszystko „do ostatniej linijki" poszło dobrze, to na końcu zwracana jest wartość 0, co oznacza sukces

```
Namaste!
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

We wczesnych standardach języka C (C89) obecność instrukcji **return** była wymagana. Od standardu C90 instrukcja ta stała się opcjonalna. Nowsze kompilatory automatycznie dołączają instrukcję **return**, jeśli nie zostanie ona napisana (Olsson, 2019).

Komentowanie kodu

Komentarze służą do wstawiania uwag do kodu źródłowego. Nie mają one wpływu na rezultat programu. Celem pisania komentarzy jest zwiększenie czytelności kodu (zarówno dla Ciebie, jak i dla innych programistów, którzy kiedyś będą mieć do czynienia z Twoim kodem). Dobra. Warto poświęcić trochę czasu na pisanie komentarzy. Czas ten odzyskasz później, gdy będziesz uruchamiał bądź modyfikował program.

Wzbogaćmy kod pierwszego programu o komentarze Dołóżmy też dodatkową linijkę tekstu oraz tajemniczy znak „\n" pojawiający się w cudzysłowach.

```c
#include <stdio.h>

int main(){
    //To jest komentarz
    printf("Namaste!\n");
    /*A to komentarz wielolinijkowy
    Za pomoca funkcji printf() drukujemy kolejna linijke tekstu:*/
    printf("Namaste to tradycyjne indyjskie przywitanie\n");
    return 0;
}

```

Zadanie 2.

1. W powyższym kodzie proszę usunąć znaczek \n z linii 5. Co się stanie ?
2. Przed słowem „to” s linijce 8. wpisać \t. Co się stanie?
3. Wyświetlić kolejną linijkę wyświetlającą wymyślony przez siebie napis.
4. Dopisać własny komentarz do tej linijki
5. Wpisać **return 0;** po pierwszej linijce drukującej tekst. Co się stanie?
6. Zakomentować linijkę z **return 0;** Co się stanie?
7. Wykasować średnik na końcu pierwszej funkcji drukującej napis na ekranie. Co się stanie? Przeanalizować treść błędu w logu.

Trochę o zmiennych

Zmienne służą do przechowywania danych podczas wykonywania programu.

Podstawowe typy zmiennych (Olsson, 2019)

Data Type	Size (Byte)	Description
char	1	Integer or character
short	2	Integer
int	4	
long	4 or 8	
long long	8	
float	4	Floating-point number
double	8	
long double	8 or 16	

https://www.tutorialspoint.com/cprogramming/c_variables.htm

Nazwa zmiennej może składać się z liter, cyfr i podkreśleń, ale nie może zaczynać się od liczby. Nie może również zawierać spacji ani znaków specjalnych. Nie może też być zastrzeżonym słowem kluczowym.

Zanim zaczniemy używać zmiennej musimy ją zadeklarować. W pierwszym przykładzie zadeklarujemy zmienną x typu całkowitego.

```c
#include <stdio.h>

int main(){
    //deklarujemy zmienna x (typu calkowitego -int)
    int x;
    //przypisujemy do x konkretna wartosc
    x=1;
    //drukujemy wartosc zmiennej wraz z komunikatem poprzedzajacym
    printf("Moja zmienna x ma wartosc:\t %d\n",x);
    return 0;
}
```

Deklaracja zmiennej (linijka 5) „mówi” kompilatorowi, że gdy napotka **x,** to ma „wiedzieć”, że jest to obiekt typu całkowitego (int).

%d oznacza, że w konkretnym miejscu w napisie trzeba wstawić zmienną całkowitoliczbową, która znajduje się w printfie po przecinku (w powyższym przypadku zmienna x).

Jeśli chcemy wyświetlać zmienne innych typów używamy innych oznaczeń. Pełną tabelkę znajdziesz np. tutaj. Po więcej informacji na ten temat zapraszam na wykład.

Zadanie 3

1. W powyższym kodzie zamiast **int x;** (linijka 5.) proszę **wpisać int x = 1;** i wykasować linijkę 7. Sprawdzić, że działanie programu nie zmieniło się (wykonaliśmy jednoczesną deklarację i inicjalizację zmiennej).
2. Zadeklarować kolejną zmienną typu całkowitego o nazwie **y** i przypisać jej wartość 3**.**
3. Wyświetlić na ekranie komunikat „Moje zmienne: x=1, y=3” .
4. Spróbuj stworzyć zmienną o nazwie **1zmienna**. Co się stanie? Przeanalizować treść błędu.

Wczytywanie danych z klawiatury

Do wczytywania danych z klawiatury służy funkcja **scanf().**

```c
#include <stdio.h>

int main(){
    //deklarujemy zmienna:
    int mojaZmienna;
    // wyswietlamy komunikat, w ktorym prosimy uzytkownika o podanie swojej zmiennej
    printf("Podaj zmienna: ");
    //wczytujemy z kla
    scanf("%d",&mojaZmienna);
    printf("Brawo, podales zmienna calkowita mojaZmienna=%d",mojaZmienna);
    return 0;
}
```

```
Podaj zmienna: 5
Brawo, podales zmienna calkowita mojaZmienna=5
```

Zadanie 4

Napisać program wczytujący wiek i numer buta użytkownika oraz wyświetlający te informacje na ekranie.

Bibliografia

Olsson, M. (2019). *Modern C Quick Syntax Reference: A Pocket Guide to the Language, APIs and Library,*. APRESS, https://doi.org/10.1007/978-1-4842-4288-9.