

# Informatyka 1 (ES1E2009)

Politechnika Białostocka - Wydział Elektryczny  
Elektrotechnika, semestr II, studia stacjonarne I stopnia  
Rok akademicki 2021/2022

## Wykład nr 1 (08.03.2022)

dr inż. Jarosław Forenc

## Dane podstawowe

- dr inż. Jarosław Forenc
- Politechnika Białostocka, Wydział Elektryczny,  
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki  
ul. Wiejska 45D, 15-351 Białystok  
WE-204
- e-mail: [j.forenc@pb.edu.pl](mailto:j.forenc@pb.edu.pl)
- tel. (0-85) 746-93-97
- <http://jforenc.prv.pl>
  - Dydaktyka - slajdy z wykładu
- konsultacje:
  - środa, 13:00-15:00, WE-204 / Teams
  - piątek, 11:00-12:00, WE-204 / Teams
  - piątek, 15:00-16:30, WE-204 / Teams (studia zaoczne)

## Program wykładu (1/2)

1. Programowanie w języku C. Deklaracje i typy zmiennych, operatory i wyrażenia arytmetyczne, operacje wejścia-wyjścia, operatory relacyjne i logiczne, wyrażenia logiczne, instrukcja warunkowa if, instrukcja switch, operator warunkowy, pętle (for, while, do .. while), tablice jednowymiarowe.
2. Informacja analogowa i cyfrowa. Pozycyjne i niepozycyjne systemy liczbowe. Konwersje pomiędzy systemami liczbowymi.
3. Jednostki informacji cyfrowej. Kodowanie informacji. Kodowanie znaków.
4. Kodowanie liczb. Reprezentacja liczb w systemach komputerowych: stałoprzecinkowa i zmiennoprzecinkowa. Standard IEEE 754.

## Program wykładu (2/2)

5. Architektura komputerów. Klasyfikacja systemów komputerowych (taksonomia Flynna). Architektura von Neumana i architektura harwardzka.
6. Budowa i zasada działania komputera. Procesor, pamięć wewnętrzna i zewnętrzna. Komunikacja z urządzeniami zewnętrznymi, interfejsy komputerowe.
7. Algorytmy. Definicja algorytmu. Klasyfikacje i sposoby przedstawiania algorytmów. Rekurencja. Złożoność obliczeniowa. Sortowanie. Klasyfikacje algorytmów sortowania. Wybrane algorytmy sortowania.
8. Zaliczenie wykładu.

## Literatura

1. S. Prata: „Język C. Szkoła programowania. Wydanie VI”. Helion, 2016.
2. R. Kawa, J. Lembas: „Wykłady z informatyki. Wstęp do informatyki”. PWN, Warszawa, 2021.
3. W. Kwiatkowski: „Wprowadzenie do kodowania”. BEL Studio, Warszawa, 2010.
4. A.S. Tanenbaum: „Strukturalna organizacja systemów komputerowych”. Helion, Gliwice, 2006.
5. K. Wojtuszkiewicz: „Urządzenia techniki komputerowej. Część 1 i 2”. PWN, Warszawa, 2013.
7. P. Wróblewski: „Algorytmy, struktury danych i techniki programowania. Wydanie VI”. Helion, Gliwice, 2019.
8. G. Coldwin: „Zrozumieć programowanie”. PWN, Warszawa, 2020.

## Efekty uczenia się i system ich oceniania

Podstawę do zaliczenia przedmiotu (uzyskanie punktów ECTS) stanowi stwierdzenie, że **każdy** z założonych **efektów uczenia się** został osiągnięty w co najmniej minimalnym akceptowalnym stopniu.

**EU1** identyfikuje i opisuje zasadę działania podstawowych elementów systemu komputerowego

**EU2** formułuje algorytmy komputerowe rozwiązujące typowe zadania inżynierskie występujące w elektrotechnice

- Szczegółowe zasady zaliczania znajdują się na stronie internetowej (<http://jforenc.prv.pl/dydaktyka.html>) oraz w systemie USOS

## Zaliczenie wykładu

- Zaliczenie odbędzie się na podstawie wyników sprawdzianu pisemnego
- Sprawdzian odbędą się w trybie bezpośrednim (na terenie Uczelni)
  - termin: **21.06.2022 (wtorek), godz. 10:15, WE-Aula II**
- Na zaliczeniu oceniane będą dwa efekty uczenia się (EU1, EU2)
- Za każdy efekt uczenia się można otrzymać od 0 do 100 pkt.
- Na podstawie otrzymanych punktów wystawiana jest ocena:

| Punkty   | Ocena | Punkty  | Ocena |
|----------|-------|---------|-------|
| 91 - 100 | 5,0   | 61 - 70 | 3,5   |
| 81 - 90  | 4,5   | 51 - 60 | 3,0   |
| 71 - 80  | 4,0   | 0 - 50  | 2,0   |

## Zaliczenie wykładu

- Każdy efekt uczenia się musi być zaliczony na ocenę pozytywną (min. 51 punktów).
- Prowadzący zajęcia może przyznawać dodatkowe punkty za aktywność na wykładzie
- Ocena końcowa wyznaczana jest na podstawie sumy otrzymanych punktów:

| Punkty    | Ocena | Punkty    | Ocena |
|-----------|-------|-----------|-------|
| 182 - 200 | 5,0   | 122 - 141 | 3,5   |
| 162 - 181 | 4,5   | 102 - 121 | 3,0   |
| 142 - 161 | 4,0   | 0 - 101   | 2,0   |

## Terminy zajęć

- Wykład nr 1 - 08.03.2022
- Wykład nr 2 - 22.03.2022
- Wykład nr 3 - 05.04.2022
- Wykład nr 4 - 12.05.2022 (czwartek)
- Wykład nr 5 - 17.05.2022
- Wykład nr 6 - 31.05.2022
- Wykład nr 7 - 14.06.2022
- Wykład nr 8 - 21.06.2022 (zaliczenie wykładu)

## Plan wykładu nr 1

- Język C
  - historia, struktura programu
  - kompilacja, zapis kodu
  - sekwencje sterujące, komentarze
  - identyfikatory (nazwy), słowa kluczowe

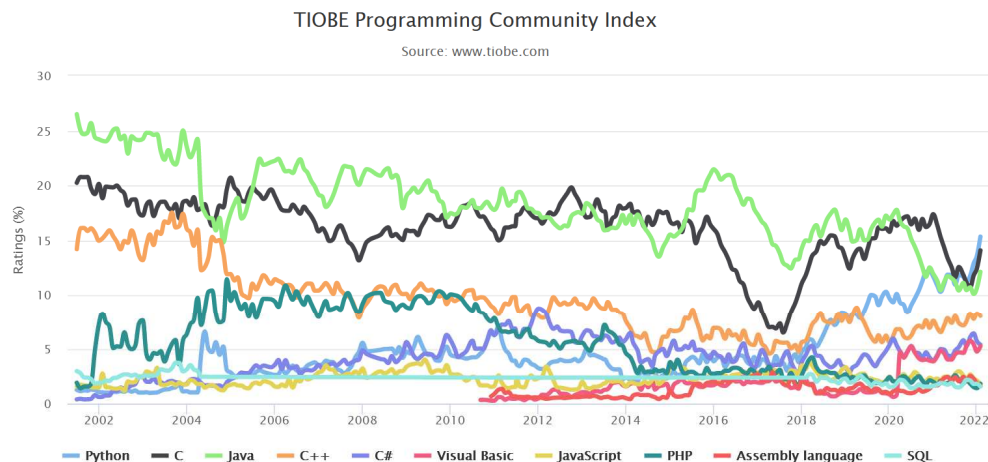
## Język C - krótka historia (1/2)

- **1969** - język BCPL - Martin Richards, University Mathematical Laboratories, Cambridge
- **1970** - język B - Ken Thompson, adaptacja języka BCPL dla pierwszej instalacji systemu Unix na komputer DEC PDP-7
- **1972** - język NB (New B), nazwany później C - Dennis Ritchie, Bell Laboratories, New Jersey, system Unix na komputerze DEC PDP-11
  - 90% kodu systemu Unix oraz większość programów działających pod jego kontrolą napisane w C
- **1978** - książka „The C Programming Language” (Kernighan, Ritchie), pierwszy podręcznik, nieformalna definicja standardu (**K&R**)

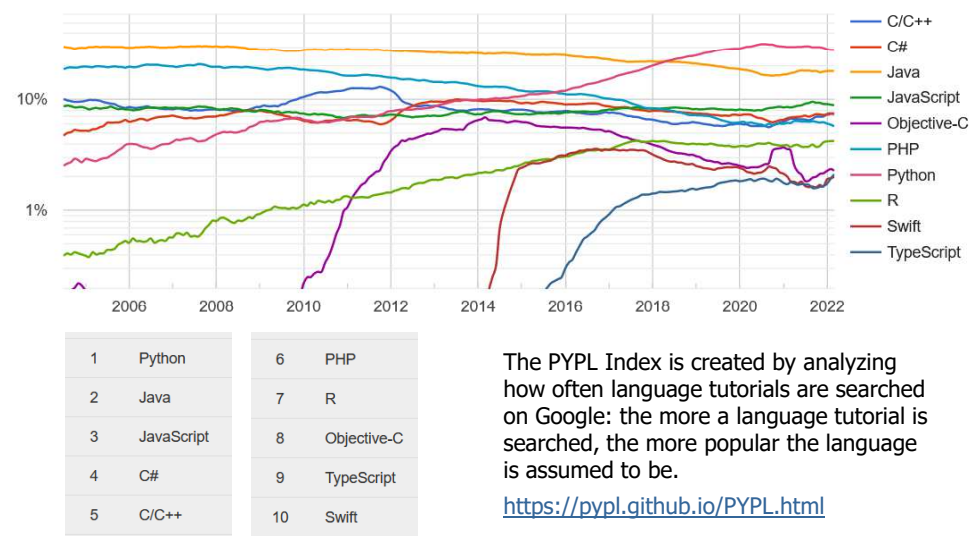
## Język C - krótka historia (2/2)

- **1989** - standard ANSI X3.159-1989 „Programming Language C” (**ANSI C, C89**)
- **1990** - adaptacja standardu ANSI C w postaci normy ISO/IEC 9899:1990 (**C90**)
- **1999** - norma ISO/IEC 9899:1999 (**C99**)
- **2011** - norma ISO/IEC 9899:2011 (**C11**)
- **2018** - norma ISO/IEC 9899:2018 (**C18** lub **C17**)

## Język C - TIOBE Programming Community Index



## Język C - PYPL Popularity of Programming Language



## Język C - pierwszy program

- Niesformatowany plik tekstowy o odpowiedniej składni i mający rozszerzenie `.c`
- Kod najprostszego programu:

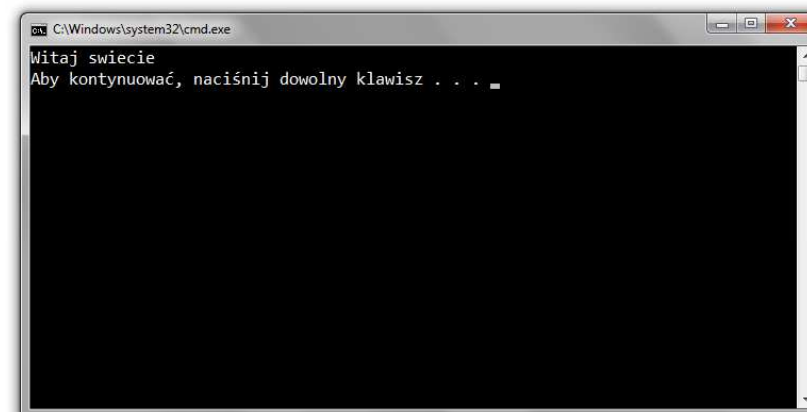
```
#include <stdio.h>

int main(void)
{
    printf("Witaj świecie\n");
    return 0;
}
```

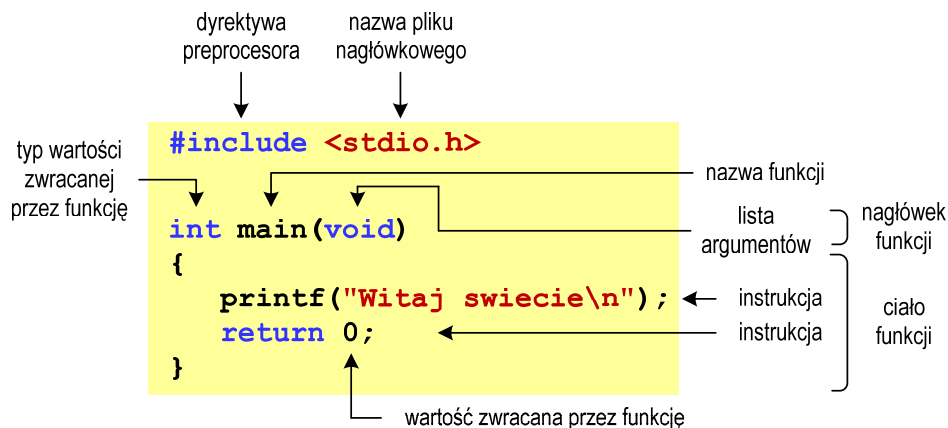
- Program konsolowy - wyświetla w konsoli tekst `Witaj świecie`

## Język C - pierwszy program

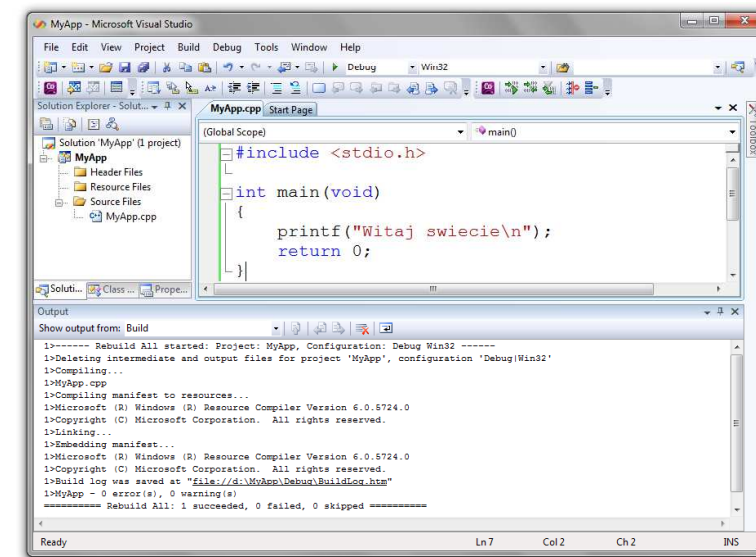
- Wynik uruchomienia programu:



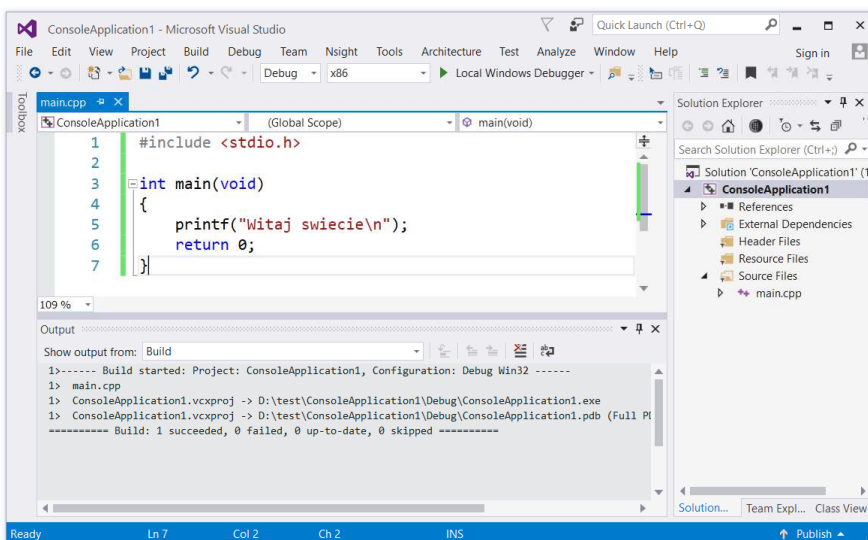
## Język C - struktura programu



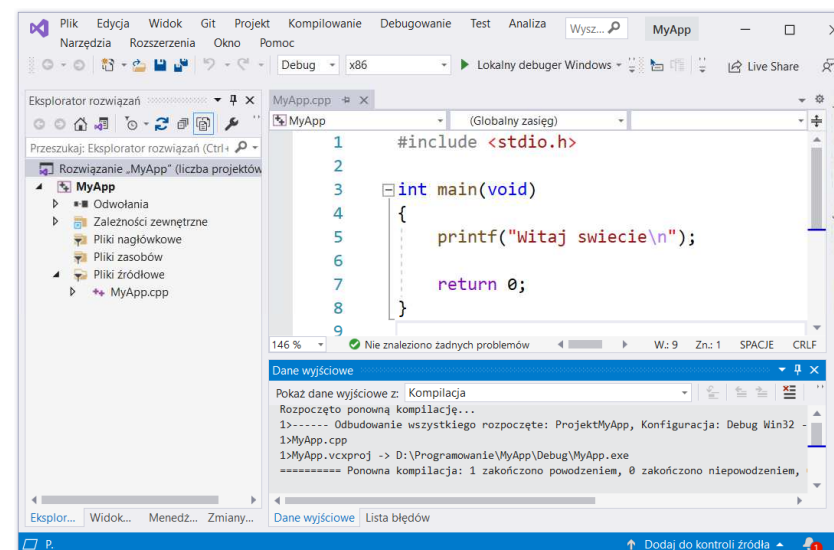
## Microsoft Visual Studio 2008



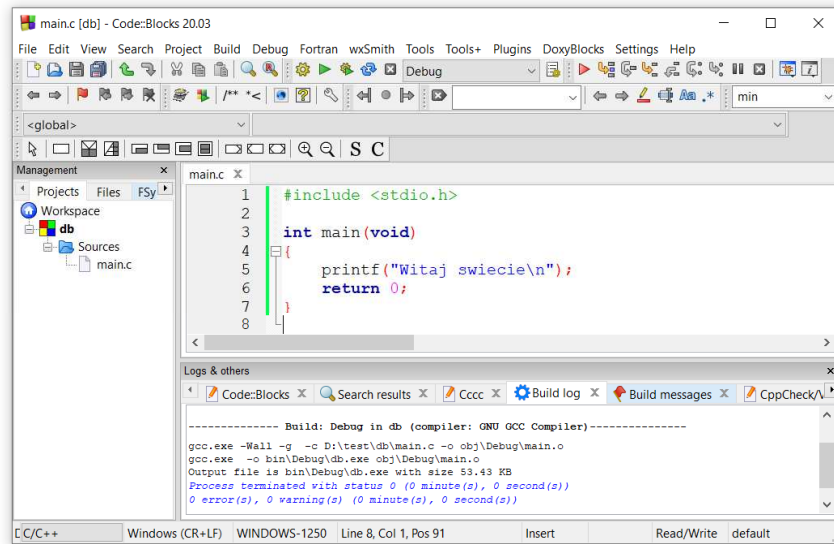
## Microsoft Visual Studio 2015



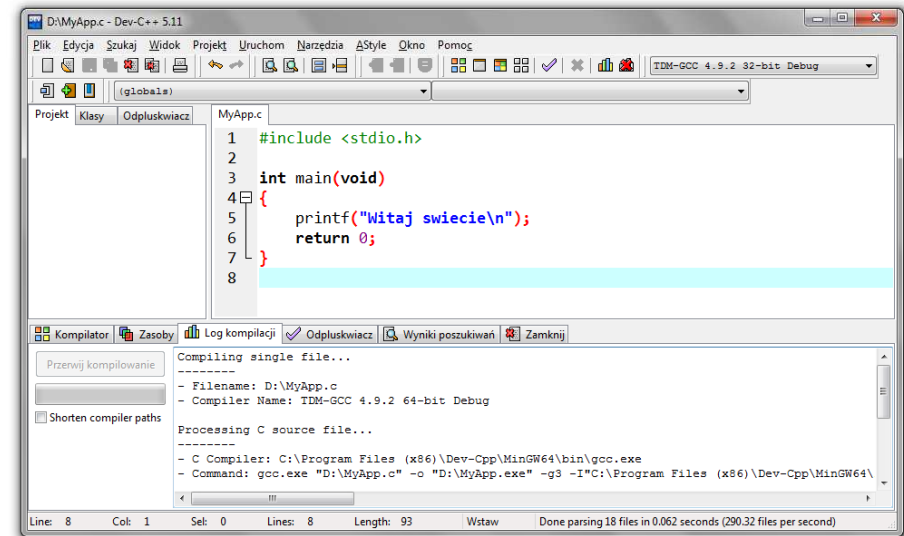
## Microsoft Visual Studio 2019



## Code::Blocks 20.03

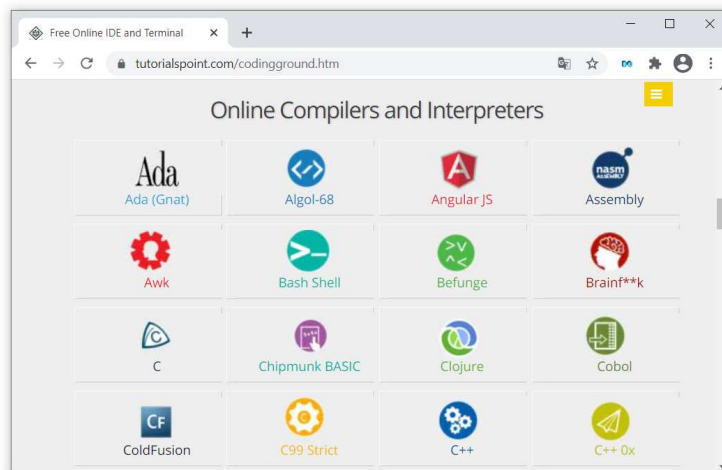


## Dev-C++ 5.11



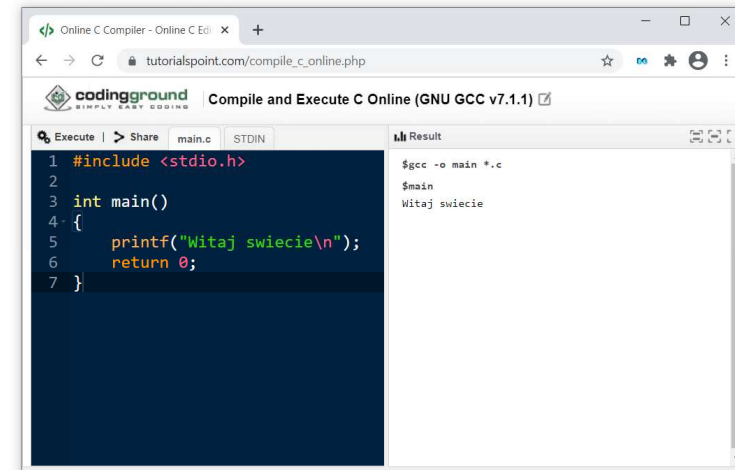
## Kompilatory on-line

- <https://www.tutorialspoint.com/codingground.htm>

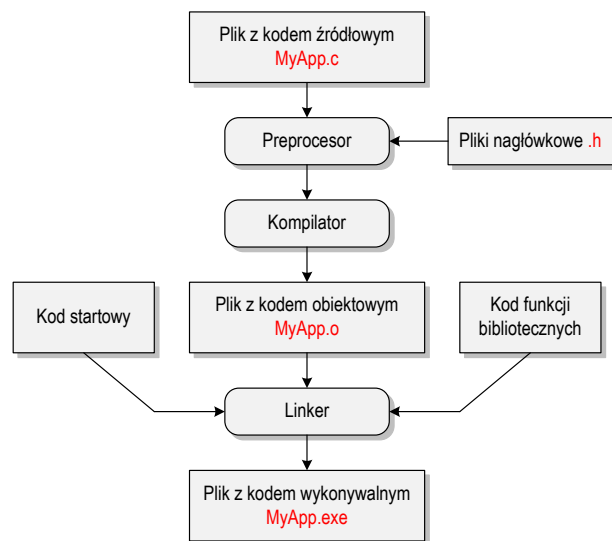


## Kompilatory on-line

- <https://www.tutorialspoint.com/codingground.htm>



## Język C - kompilacja programu



## Język C - Wyświetlanie tekstu (printf)

- Znak przejścia do nowego wiersza `\n` może pojawić w dowolnym miejscu łańcucha znaków

```
printf("Witaj swiecie\n");
```

Witaj swiecie  
—

```
printf("Witaj\nswiecie\n");
```

Witaj  
swiecie  
—

```
printf("Witaj ");  
printf("swiecie");  
printf("\n");
```

Witaj swiecie  
—

## Język C - zapis kodu programu

- Sposób zapisu kodu programu wpływa tylko na jego przejrzystość, a nie na kompilację i wykonanie
- W takiej postaci program także skompiluje się:

```
#include <stdio.h>  
int main(void) {printf("Witaj swiecie\n");return 0;}
```

- W Microsoft Visual Studio 2008 można automatycznie sformatować kod źródłowy programu - **Ctrl + K + F**
- Język C rozróżnia wielkość liter - poniższy kod nie skompiluje się:

```
#include <stdio.h>  
int Main(void) {printf("Witaj swiecie\n");return 0;}
```

## Język C - Sekwencje sterujące

- Istnieją także inne sekwencje sterujące (ang. escape sequence)

| Opis znaku  | Zapis w printf() |
|---|------------------|
| Alarm (ang. alert), głośniczek wydaje dźwięk      | <code>\a</code>  |
| Backspace   | <code>\b</code>  |
| Wysunięcie strony (ang. form feed)                | <code>\f</code>  |
| Przejście do nowego wiersza (ang. new line)       | <code>\n</code>  |
| CR - Carriage Return (powrót na początek wiersza) | <code>\r</code>  |
| Tabulacja pozioma (odstęp) (ang. horizontal tab)  | <code>\t</code>  |
| Tabulacja pionowa (ang. vertical tab)             | <code>\v</code>  |

## Język C - Wyświetlenie znaków specjalnych

- Niektóre znaki pełnią specjalną funkcję i nie można wyświetlić ich w tradycyjny sposób

| Opis znaku               | Znak | Zapis w printf() |
|--------------------------|------|------------------|
| Cudzysłów                | "    | \"               |
| Apostrof                 | '    | \'               |
| Ukośnik (ang. backslash) | \    | \\               |
| Procent                  | %    | %%               |

Ścieżka dostępu: "C:\dane\plik.txt"

```
printf("Ścieżka dostępu: \"C:\\dane\\plik.txt\\n");
```

## Język C - Wyświetlenie tekstu

```
#include <stdio.h>

int main(void)
{
    printf("-----\n");
    printf("| Punkty | Ocena |\n");
    printf("-----\n");
    printf("| 91-100 | 5,0 |\n");
    printf("| 81-90 | 4,5 |\n");
    printf("| 71-80 | 4,0 |\n");
    printf("| 61-70 | 3,5 |\n");
    printf("| 51-60 | 3,0 |\n");
    printf("| 0-50 | 2,0 |\n");
    printf("-----\n");

    return 0;
}
```

| Punkty | Ocena |
|--------|-------|
| 91-100 | 5,0   |
| 81-90  | 4,5   |
| 71-80  | 4,0   |
| 61-70  | 3,5   |
| 51-60  | 3,0   |
| 0-50   | 2,0   |

## Język C - Wyświetlenie znaku o podanym kodzie

- Można wyświetlić dowolny znak podając jego kod w systemie ósemkowym lub szesnastkowym

| Znaczenie   | Zapis |
|---|-------|
| Znak o podanym kodzie ASCII (system ósemkowy)     | \ooo  |
| Znak o podanym kodzie ASCII (system szesnastkowy) | \xhh  |

```
printf("\127\151\164\141\152\040");
printf("\x73\x77\x69\x65\x63\x69\x65\x21\x0A");
```

Witaj świecie!

## Język C - Komentarze

- Komentarze są pomijane podczas kompilacji

```
/*
   Nazwa: MyApp.c
   Autor: Jarosław Forenc, Politechnika Białostocka
   Data: 04-03-2022 17:00
   Opis: Program wyświetlający tekst "Witaj świecie"
*/

#include <stdio.h> // zawiera deklarację printf()

int main(void) // nagłówek funkcji main()
{
    printf/* funkcja */("Witaj świecie\n");

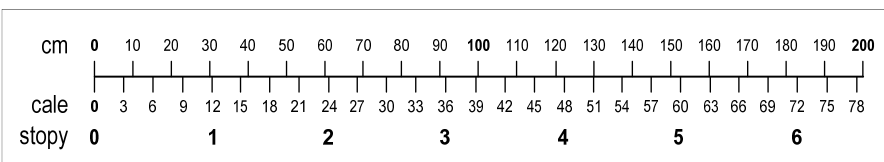
    return 0;
}
```



## Przykład: zamiana wzrostu w cm na stopy i cale

### Wybrane jednostki długości w brytyjskim systemie miar:

- 1 cal (inch) [in] = 2,54 [cm]
- 1 stopa (foot) [ft] = 12 cali = 30,48 [cm]



- 1 jard (yard) [yd] = 3 stopy = 91,44 [cm]
- 1 furlong [fur] = 660 stóp = 201,168 [m]
- 1 mila (mile) [mi] = 8 furlongów = 1609,344 [m]

## Język C - identyfikatory (nazwy)

- Dozwolone znaki: **A-Z, a-z, 0-9, \_** (podkreślenie)
- Długość nie jest ograniczona (rozdzielalne są 63 pierwsze znaki)
- Poprawne identyfikatory:

```
temp    u2    u_2    pole_kola    alfa    Beta    XyZ
```

- Pierwszym znakiem nie może być cyfra
- W identyfikatorach nie można stosować spacji, liter diakrytycznych
- Błędne identyfikatory:

```
2u    pole kola    pole_koła
```

## Przykład: zamiana wzrostu w cm na stopy i cale

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    float cm;    /* wzrost w cm */
```

```
    float stopy; /* wzrost w stopach */
```

```
    float cale;  /* wzrost w calach */
```

```
    printf("Podaj wzrost w cm: ");
```

```
    scanf("%f", &cm);
```

```
    stopy = cm / 30.48f;
```

```
    cale = cm / 2.54f;
```

```
    printf("%f [cm] = %f [ft]\n", cm, stopy);
```

```
    printf("%f [cm] = %f [in]\n", cm, cale);
```

```
    return 0;
```

```
}
```

```
Podaj wzrost w cm: 175
175.000000 [cm] = 5.741470 [ft]
175.000000 [cm] = 68.897636 [in]
```

## Język C - identyfikatory (nazwy)

- Nie zaleca się, aby pierwszym znakiem było podkreślenie
- Identyfikatory nie powinny być zbyt długie

```
_temp    __temp    temperatura_w_skali_Celsiusza
```

- Nazwa **zmiennej** powinna być związana z jej zawartością
- Język C rozróżnia wielkość liter więc poniższe zapisy oznaczają inne identyfikatory

```
tempc    Tempc    TempC    TEMPC    TeMpC
```

- Jako nazw zmiennych nie można stosować **słów kluczowych** języka C

## Język C - słowa kluczowe języka C

- W standardzie C11 zdefiniowane są 43 słowa kluczowe

|                       |                       |                       |                             |
|-----------------------|-----------------------|-----------------------|-----------------------------|
| <code>auto</code>     | <code>extern</code>   | <code>short</code>    | <code>while</code>          |
| <code>break</code>    | <code>float</code>    | <code>signed</code>   | <code>_Alignas</code>       |
| <code>case</code>     | <code>for</code>      | <code>sizeof</code>   | <code>_Alignof</code>       |
| <code>char</code>     | <code>goto</code>     | <code>static</code>   | <code>_Bool</code>          |
| <code>const</code>    | <code>if</code>       | <code>struct</code>   | <code>_Complex</code>       |
| <code>continue</code> | <code>inline</code>   | <code>switch</code>   | <code>_Generic</code>       |
| <code>default</code>  | <code>int</code>      | <code>typedef</code>  | <code>_Imaginary</code>     |
| <code>do</code>       | <code>long</code>     | <code>union</code>    | <code>_Noreturn</code>      |
| <code>double</code>   | <code>register</code> | <code>unsigned</code> | <code>_Static_assert</code> |
| <code>else</code>     | <code>restrict</code> | <code>void</code>     | <code>_Thread_local</code>  |
| <code>enum</code>     | <code>return</code>   | <code>volatile</code> |                             |

## Koniec wykładu nr 1

Dziękuję za uwagę!