

# Programowanie w języku C (EAR1S02005)

---

Politechnika Białostocka - Wydział Elektryczny  
Automatyka i Robotyka, semestr II, studia stacjonarne I stopnia  
Rok akademicki 2020/2021

**Zajęcia nr 1 (23.02.2021)**

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 - dodatkowe materiały do zajęć
- konsultacje:
  - wtorek, godz. 10:30-12:00, WE-204 / Teams
  - piątek, godz. 12:30-14:00, WE-204 / Teams
  - piątek, godz. 17:00-18:30, WE-204 / Teams (studia zaoczne)
  - niedziela, godz. 08:00-09:00, Teams (studia zaoczne)

## Program przedmiotu (1/2)

L.p.	Temat	Godz.
1.	Zajęcia organizacyjne. Obsługa wybranego środowiska przeznaczonego do tworzenia, analizy i uruchamiania programów w języku C. Ogólna struktura programu w języku C.	2
2.	Operacje wejścia-wyjścia w języku C. Zmienne, typy zmiennych. Operatory i wyrażenia arytmetyczne. Priorytet operatorów. Funkcje matematyczne.	2
3.	Wydanie tematów projektów. Operatory relacyjne i logiczne. Instrukcja warunkowa if. Operator warunkowy ? :. Instrukcja switch.	2
4.	Instrukcje iteracyjne for, while i do ... while.	2

## Program przedmiotu (2/2)

L.p.	Temat	Godz.
5.	Tablice jedno- i dwuwymiarowe. Łańcuchy znaków.	2
6.	Struktury.	2
7.	Wskaźniki. Dynamiczny przydział pamięci w języku C.	2
8.	Funkcje. Przekazywanie argumentów do funkcji przez wartość i wskaźnik. Rekurencja.	2
9.	Pliki tekstowe i binarne w języku C.	2
10.	Budowa programów z wykorzystaniem elementów języka C.	10
11.	Zaliczenie zajęć - przedstawienie i ocena projektów.	2

## Literatura

1. S. Prata: Język C. Szkoła programowania. Wydanie VI. Helion, 2016.
2. B.W. Kernighan, D.M. Ritchie: Język ANSI C. Programowanie. Wydanie II. Helion, Gliwice, 2010.
3. P.J. Deitel, H. Deitel: Język C. Solidna wiedza w praktyce. Wydanie VIII. Helion, Gliwice, 2020.
4. K.N. King: Język C. Nowoczesne programowanie. Wydanie II. Helion, Gliwice, 2011.
5. S.G. Kochan: Język C. Kompendium wiedzy. Wydanie IV. Helion, 2015.
6. R. Reese: Wskaźniki w języku C. Przewodnik. Helion, Gliwice, 2014.

## Warunki zaliczenia przedmiotu

- Obecność na zajęciach (więcej niż dwie nieusprawiedliwione nieobecności skutkują niezaliczeniem projektu)
- Realizacja w trakcie zajęć zadań przedstawionych przez prowadzącego
- Zaliczenie wszystkich sprawozdań w postaci prac domowych, co stanowi: 1/3 oceny końcowej. Sprawozdania będą zawierały zadania podzielone na oceny 3.0, 4.0 oraz 5.0. W semestrze przewiduje się 4-5 sprawozdań.
- Zaliczenie projektu końcowego. Ocena z projektu stanowi 2/3 oceny końcowej.
- Prowadzący zajęcia może przyznawać dodatkowe bonusy za aktywność na zajęciach.
- Ocena końcowa wyznaczana jest na podstawie sumy algebraicznej z ocen (projekt, prace domowe oraz bonusy).

## 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.

<b>EU3</b>	potrafi napisać program w języku C w oparciu o przygotowane założenia
<b>EU4</b>	potrafi stosować właściwe techniki programistyczne do realizacji programu
<b>EU5</b>	potrafi przetestować program oraz wykryć i zneutralizować przyczynę złego funkcjonowania programu

# System oceniania

- Student, który zalicza na ocenę **dostateczny (3)** powinien:
  - umieć w wybranym środowisku programistycznym: stworzyć projekt, wprowadzić kod źródłowy programu w języku C, skompilować go i uruchomić
  - właściwie interpretować błędy i ostrzeżenia kompilacji oraz odpowiednio poprawiać kod programu
  - znać ogólną strukturę programu w języku C i pisać programy oparte na tej strukturze
  - umieć opracować algorytm rozwiązujący prosty problem inżynierski i zaimplementować go w postaci programu
  - umieć deklarować zmienne właściwie dobierając ich typ
  - obliczać proste wyrażenia arytmetyczne stosując odpowiednie operatory i funkcje matematyczne
  - umieć wyświetlać dane (funkcja **printf()**) oraz je wczytywać (funkcja **scanf()**) stosując odpowiednie specyfikatory formatu



# System oceniania

- Student, który zalicza na ocenę **dostateczny (3)** powinien (c.d.):
  - umieć stosować komentarze opisujące kod programu
  - umieć budować proste wyrażenia logiczne zawierające operatory logiczne, relacyjne (porównania) i arytmetyczne
  - umieć sterować wykonaniem programu stosując proste instrukcje warunkowe if-else
  - umieć sterować wykonaniem programu stosując zagnieżdżone pętle
  - umieć wykonywać podstawowe operacje na tablicach jednowymiarowych (wektorach)
  - umieć pisać proste funkcje rozwiązujące typowe zadania inżynierskie występujące w elektrotechnice
  - umieć odczytać dane z pliku tekstowego i zapisać wyniki do pliku tekstowego

# System oceniania

- Student, który zalicza na ocenę **dobry (4)** powinien (oprócz wymagań na ocenę 3):
  - umieć budować złożone wyrażenia logiczne zawierające operatory logiczne, relacyjne (porównania) i arytmetyczne
  - wykonywać elementarne operacje na tablicach dwuwymiarowych (macierzach)
  - umieć pisać złożone funkcje z przekazywaniem argumentów przez wartość
  - umieć odczytać dane z pliku binarnego i zapisać wyniki do pliku binarnego

# System oceniania

- Student, który zalicza na ocenę **bardzo dobry (5)** powinien (oprócz wymagań na ocenę 4):
  - umieć wykorzystać dynamiczny przydział pamięci
  - umieć pisać złożone funkcje z przekazywaniem argumentów przez wskaźnik

# Język C - pierwszy program

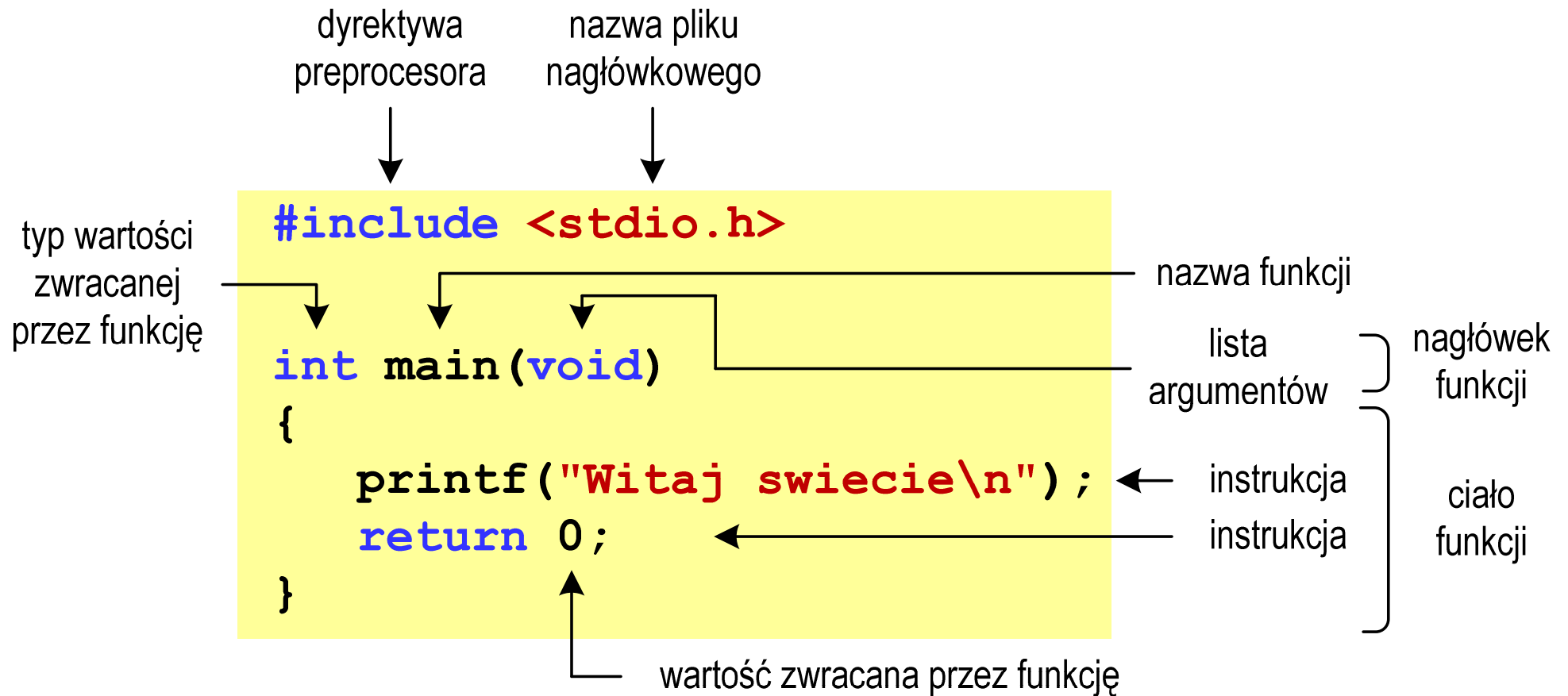
- Niesformatowany plik tekstowy o odpowiedniej składni i mający rozszerzenie `.c`
- Kod najprostszego programu:

```
#include <stdio.h>

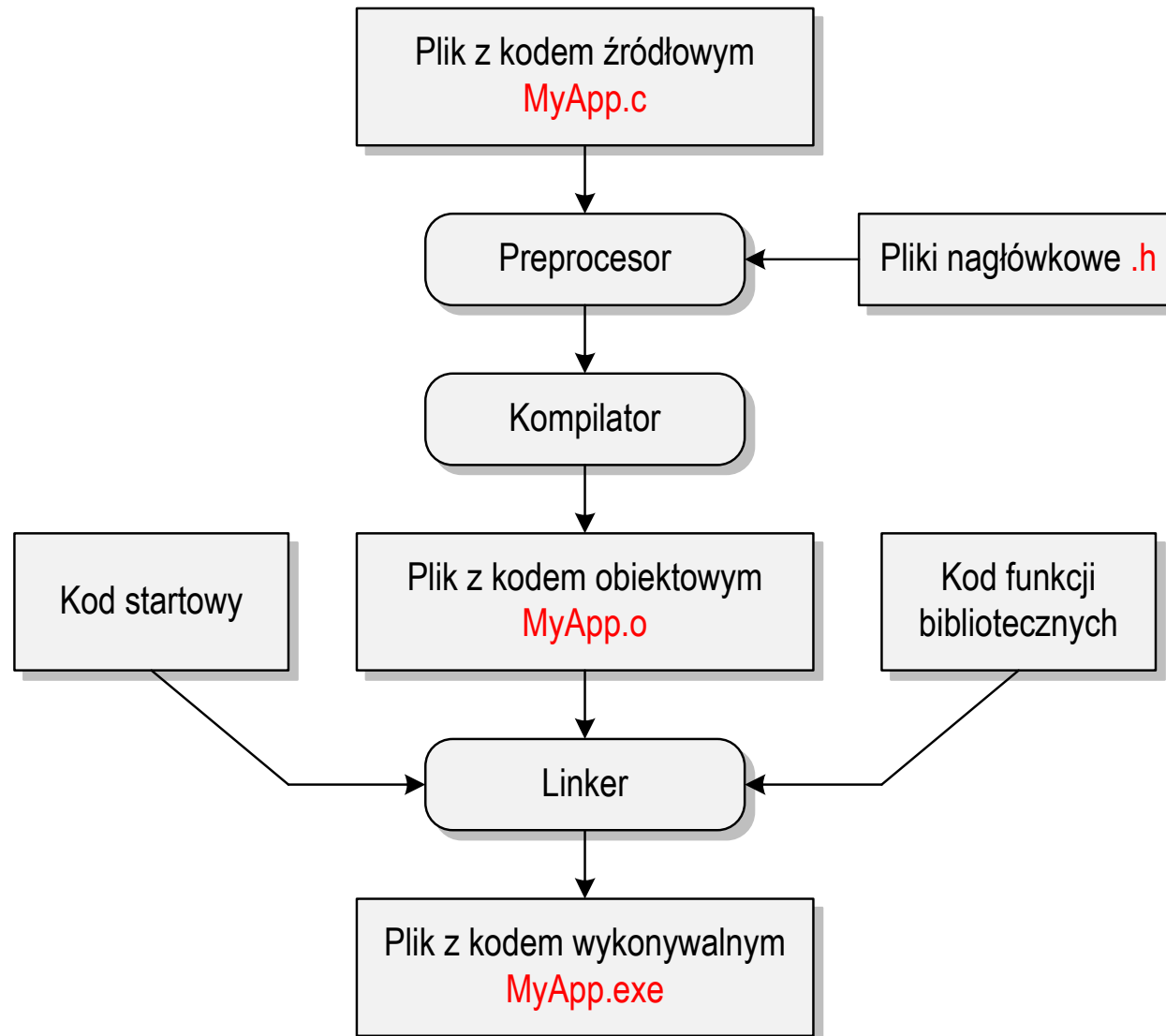
int main(void)
{
    printf("Witaj swiecie\n");
    return 0;
}
```

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

# Język C - struktura programu



# Język C - kompilacja programu



## 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;}
```

- Język C rozróżnia **wielkość liter** - poniższy kod nie skompiluje się:

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

## 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 - Sekwencje sterujące

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

<b>Opis znaku</b>	<b>Zapis w printf()</b>
Alarm (ang. alert), głośniczek wydaje dźwięk	<code>\a</code>
Backspace	<code>\b</code>
Wysunięcie strony (ang. form feed)	<code>\f</code>
Przejsie 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	%	%%

```
Sciezka dostepu: "C:\dane\plik.txt"
```

```
printf("Sciezka dostepu: \"C:\\dane\\plik.txt\\\"\\n");
```

## 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)	<code>\0oo</code>
Znak o podanym kodzie ASCII (system szesnastkowy)	<code>\xhh</code>

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

```
Witaj swiecie!
```

## 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 - Komentarze

- Komentarze są pomijane podczas kompilacji

```
/*  
  Nazwa: MyApp.c  
  Autor: Jarosław Forenc, Politechnika Białostocka  
  Data: 23-02-2021 12:15  
  Opis: Program wyświetlający tekst "Witaj świecie"  
*/  
  
#include <stdio.h>      // zawiera deklarację printf()  
  
int main(void)          // nagłówek funkcji main()  

```