

## INFORMATYKA - Pracownia specjalistyczna 30 h

Kod przedmiotu: EDS1B1007, ECTS: 5 pkt.

Kierunek: Elektrotechnika, studia stacjonarne I stopnia, dualne

Semestr: I, rok akademicki: 2019/2020

Grupa: PS1, poniedziałek, godz. 14:15-15:45, WE-110

dr inż. Jarosław Forenc

WE-204, tel. (0-85) 746-93-97

e-mail: [j.forenc@pb.edu.pl](mailto:j.forenc@pb.edu.pl)

WWW: <http://jforenc.prv.pl/> (Dydaktyka)

Konsultacje: poniedziałek, godz. 10:00-11:30, WE-204

piątek, godz. 10:00-11:30, WE-204

sobota, godz. 14:30-16:00, WE-204 (zaoczne)

### Program szczegółowy:

- Zajęcia organizacyjne. Obsługa wybranego środowiska przeznaczonego do tworzenia, analizy i uruchamiania programów w języku C. Kompilacja i konsolidacja programów. Ogólna struktura programu w języku C. Komentarze.
- Operacje wejścia-wyjścia w języku C. Zmienne, deklaracje, typy i nazwy zmiennych. Stałe liczbowe. Operatory i wyrażenia arytmetyczne. Priorytet operatorów. Funkcje matematyczne.
- Operatory relacyjne (porównania) i logiczne. Wyrażenia logiczne. Instrukcja warunkowa if. Zagnieżdżanie if-else. Operator warunkowy ? :. Instrukcja wyboru wielowariantowego switch.
- Instrukcja iteracyjna for. Zagnieżdżanie pętli for. Instrukcje break, goto i continue. Instrukcje iteracyjne while i do ... while.
- Tablice jednowymiarowe, deklaracja tablicy jednowymiarowej, inicjalizacja elementów tablicy, operacje na tablicach jednowymiarowych.
- Tablice dwu- i wielowymiarowe, operacje na tablicach.
- Łańcuchy znaków. Plik nagłówkowy string.h.
- Kolokwium nr 1.
- Struktury, odwołania do pól struktury. Inicjalizacja zmiennej strukturalnej.
- Wskaźniki. Dynamiczny przydział pamięci w języku C.
- Funkcje, ogólna struktura funkcji. Umieszczanie definicji funkcji w programie.
- Funkcje, przekazywanie argumentów do funkcji przez wartość i wskaźnik. Rekurencyjne wywołanie funkcji.
- Operacje wejścia-wyjścia, pliki tekstowe w języku C.
- Pliki binarne w języku C.
- Kolokwium nr 2.

### Literatura:

- S. Prata: *Język C. Szkoła programowania*. Wydanie VI. Helion, Gliwice, 2016.
- B.W. Kernighan, D.M. Ritchie: *Język ANSI C. Programowanie*. Wyd. II. Helion, Gliwice, 2010.
- R. Wileczek: *Microsoft Visual C++ 2008. Tworzenie aplikacji dla Windows*. Helion, Gliwice, 2009.

4. S.G. Kochan: *Język C. Kompendium wiedzy*. Wydanie IV. Helion, Gliwice, 2015.

5. R. Reese: *Wskaźniki w języku C. Przewodnik*. Helion, Gliwice, 2014.

6. Instrukcje do pracowni: <https://we.pb.edu.pl/ketim/materialy-dydaktyczne-ketim/>

7. Materiały na stronie internetowej: <http://jforenc.prv.pl/dydaktyka.html>

### Warunki zaliczenia przedmiotu:

- Obecność na zajęciach:
  - więcej niż trzy nieusprawiedliwione nieobecności skutkują niezaliczeniem pracowni.
- Realizacja w trakcie zajęć zadań przedstawionych przez prowadzącego.
- Zaliczenie dwóch kolokwium - praktycznych sprawdzianów pisania programów komputerowych:
  - kolokwia odbędą się na zajęciach nr 8 (25.11.2019) i 15 (27.01.2020),
  - poprawy kolokwium odbędą się poza zajęciami,
  - na kolokwiach można korzystać z materiałów w formie drukowanej (instrukcje do przedmiotu, książki, notatki, itp.),
  - za każde kolokwium 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

- oba kolokwia muszą być zaliczone na ocenę pozytywną (3,0 lub wyżej).
- Zaliczenie prac domowych i kartkówek:
    - prace domowe polegają na napisaniu programów komputerowych wskazanych przez prowadzącego zajęcia,
    - pracę domową należy wysłać na adres e-mailowy: [j.forenc@pb.edu.pl](mailto:j.forenc@pb.edu.pl) do końca dnia (godz. 23:59), w którym odbywają się kolejne zajęcia,
    - błędy w programach mogą być poprawione do końca dnia, w którym będą odbywały się następne zajęcia,
    - za każdy poprawnie działający program student otrzymuje 1 pkt.,
    - w przypadku stwierdzenia niesamodzielności pracy domowej: nie jest ona zaliczana, nie można jej ponownie oddać, student otrzymuje -1 pkt.,
    - należy wysłać tylko pliki z kodem źródłowym (.cpp, .c),
    - przed wysłaniem programu należy sprawdzić czy program:
      - kompile się oraz kompilator nie wyświetla ostrzeżeń,
      - jest poprawnym rozwiązaniem zadania,
      - ma odpowiednio sformatowany kod źródłowy (formatowanie kodu: Ctrl + K + F),
      - nosi nazwę zgodną z poniższym wzorcem:  
Nazwisko\_Imie\_NrInstrukcji\_NrZadania.cpp np.  
Kowalski\_Jan\_INF\_D02\_3.cpp Nowak\_Anna\_INF\_D11\_5.cpp

- zawiera na początku komentarz nagłówkowy:

```
/*
Nazwa: Kowalski_Jan_INF_D02_3.cpp
Autor: Jan Kowalski
Album: 123456
Data: 01-10-2019
Kod: EDS1B1007
Forma: Pracownia specjalistyczna
IDE: Microsoft Visual Studio 2008
Opis: Program wyświetlający tekst "Witaj świecie"
*/
```

- kartkówki odbędą się na wybranych zajęciach,
- kartkówek nie poprawiamy,
- co najmniej 70% prac domowych i kartkówek musi być zaliczonych.

5. Ocena końcowa wyznaczana jest na podstawie sumy otrzymanych punktów za kolokwia:

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

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

### Efekty uczenia się i system ich oceniania:

Student, który zaliczył przedmiot:

EU3	pisze i uruchamia programy strukturalne w języku C stosując odpowiednie typy, instrukcje warunkowe, pętle i tablice
<p>Student, który zalicza na ocenę <b>dostateczny (3,0)</b> powinien:</p> <ul style="list-style-type: none"> <li>- umieć w wybranym środowisku programistycznym: stworzyć projekt, wprowadzić kod źródłowy programu w języku C, skompilować go i uruchomić;</li> <li>- właściwie interpretować błędy i ostrzeżenia kompilacji oraz odpowiednio poprawiać kod programu;</li> <li>- znać ogólną strukturę programu w języku C i pisać programy oparte na tej strukturze;</li> <li>- umieć opracować algorytm rozwiązujący prosty problem inżynierski i zaimplementować go w postaci programu;</li> <li>- umieć deklarować zmienne właściwie dobierając ich typ;</li> <li>- obliczać proste wyrażenia arytmetyczne stosując odpowiednie operatory i funkcje matematyczne;</li> <li>- umieć wyświetlać dane (funkcja <b>printf()</b>) oraz je wczytywać (funkcja <b>scanf()</b>) stosując odpowiednie specyfikatory formatu;</li> <li>- umieć stosować komentarze opisujące kod programu;</li> <li>- umieć budować proste wyrażenia logiczne zawierające operatory logiczne, relacyjne (porównania) i arytmetyczne;</li> <li>- umieć sterować wykonaniem programu stosując proste instrukcje warunkowe <b>if-else</b>, pętle <b>for</b>, <b>while</b> i <b>do..while</b>;</li> </ul>	

- umieć definiować tablice jednowymiarowe (wektory) i tablice dwuwymiarowe (macierze) oraz wykonywać podstawowe operacje na tych tablicach;
- znać sposób przechowywania i inicjalizacji łańcuchów znaków (napisów), umieć stosować funkcje z pliku nagłówkowego **string.h**.

Student, który zalicza na ocenę **dobry (4,0)** powinien (oprócz wymagań na ocenę 3):

- umieć budować złożone wyrażenia logiczne zawierające operatory logiczne, relacyjne (porównania) i arytmetyczne;
- umieć sterować wykonaniem programu stosując zagnieżdżone instrukcje warunkowe **if-else**, instrukcję wyboru wielowariantowego **switch** oraz zagnieżdżone pętle **for**;
- umieć wykonywać zaawansowane operacje na tablicach jednowymiarowych.

Student, który zalicza na ocenę **bardzo dobry (5,0)** powinien (oprócz wymagań na ocenę 4):

- umieć zastosować w kodzie programu operator warunkowy zastępujący instrukcję **if-else**;
- umieć zastosować instrukcje **break** i **continue** w pętlach **for**, **while** i **do..while**;
- umieć wykonywać zaawansowane operacje na tablicach dwuwymiarowych.

EU4	definiuje i wykorzystuje własne funkcje oraz stosuje operacje zapisu i odczytu plików w samodzielnie napisanych programach komputerowych w języku C
<p>Student, który zalicza na ocenę <b>dostateczny (3,0)</b> powinien:</p> <ul style="list-style-type: none"> <li>- znać ogólną strukturę funkcji w języku C, definiować własne funkcje zgodnie z tą strukturą, rozumieć przebieg wykonania programu zawierającego wywołania takich funkcji;</li> <li>- umieć definiować funkcje bezargumentowe lub z argumentami, zwracające lub niezwracające wartości oraz odpowiednio je wywoływać;</li> <li>- znać i stosować proste struktury;</li> <li>- umieć przydzielić dynamicznie i zwolnić pamięć na tablicę jednowymiarową (wektor);</li> <li>- znać schemat przetwarzania pliku i potrafić zastosować go w samodzielnie napisanych programach;</li> <li>- umieć otworzyć plik w trybie tekstowym lub binarnym do zapisu lub odczytu oraz zamknąć ten plik;</li> <li>- umieć pisać proste programy przetwarzające pliki tekstowe i binarne zawierające liczby całkowite i/lub rzeczywiste;</li> <li>- znać i stosować w programach podstawowe funkcje do odczytu i zapisu plików tekstowych (<b>fscanf</b>, <b>fprintf</b>) oraz binarnych (<b>fread</b>, <b>fwrite</b>).</li> </ul> <p>Student, który zalicza na ocenę <b>dobry (4,0)</b> powinien (oprócz wymagań na ocenę 3):</p> <ul style="list-style-type: none"> <li>- rozróżniać pojęcia deklaracji i definicji funkcji, stosować we właściwy sposób prototypy funkcji;</li> <li>- umieć stosować złożone struktury;</li> <li>- umieć przydzielić dynamicznie i zwolnić pamięć na tablicę dwuwymiarową (macierz);</li> <li>- umieć pisać programy przetwarzające pliki tekstowe i binarne zawierające dane różnych typów prostych i złożonych;</li> <li>- znać i stosować w programach funkcje znakowe i łańcuchowe do odczytu i zapisu plików tekstowych oraz funkcje rekordowe (blokowe) do operacji na plikach binarnych.</li> </ul> <p>Student, który zalicza na ocenę <b>bardzo dobry (5,0)</b> powinien (oprócz wymagań na ocenę 4):</p> <ul style="list-style-type: none"> <li>- znać różnice w przekazywaniu argumentów do funkcji przez wartość i wskaźnik oraz odpowiednio stosować obie metody w programach;</li> <li>- umieć definiować własne funkcje rekurencyjne i stosować je w programach;</li> <li>- umieć pisać programy przetwarzające jednocześnie wiele plików tekstowych i binarnych.</li> </ul>	

## Wymagania BHP:

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

1. Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
2. Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
3. Sprawdzić prawidłowość połączeń urządzeń.
4. Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
5. W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.
6. W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
7. Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
8. Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.
9. W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
10. Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
11. Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
12. W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.