

# POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

## KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2020/2021

Wydział Informatyki i Telekomunikacji

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: stacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

### 1 INFORMACJE O PRZEDMIOCIE

|                                         |                                      |
|-----------------------------------------|--------------------------------------|
| NAZWA PRZEDMIOTU                        | Języki i paradygmaty programowania I |
| NAZWA PRZEDMIOTU<br>W JĘZYKU ANGIELSKIM | Programming languages and paradigms  |
| KOD PRZEDMIOTU                          | WiIT I oIS C11 20/21                 |
| KATEGORIA PRZEDMIOTU                    | Przedmioty kierunkowe                |
| LICZBA PUNKTÓW ECTS                     | 4.00                                 |
| SEMESTRY                                | 2                                    |

### 2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

| SEMESTR | WYKŁAD | ĆWICZENIA | LABORATORIUM | LABORATORIUM<br>KOMPUTERO-<br>WE | SEMINARIUM | PROJEKT |
|---------|--------|-----------|--------------|----------------------------------|------------|---------|
| 2       | 30     | 0         | 30           | 0                                | 0          | 0       |

### 3 CELE PRZEDMIOTU

**Cel 1** Poznanie podstawowych paradygmatów programowania, w tym paradygmatu proceduralnego oraz obiektowego. Osiągnięcia umiejętności w ocenie przydatności paradygmatów do rozwiązywania różnego typu problemów; projektowania, implementacji, testowania i debugowania prostych programów proceduralnych.

**Cel 2** Poznanie składni języka C jako przykładu języka strukturalnego pozwalającego na programowanie proceduralne

**Cel 3** Podstawy proceduralnego podejścia do programowania - tworzenie funkcji oraz procedur, które potrafią obsługiwać dowolne typy danych.

**Cel 4** Zamykanie danych w oddzielnych plikach, udostępnianie działań nad danymi przez funkcje - elementy stylu obiektowego w języku C.

## 4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 zaliczenie przedmiotu: Wstęp do programowania

## 5 EFEKTY KSZTAŁCENIA

**EK1 Wiedza** Student rozumie pojęcie paradygmat programowania. Ma wiedzę ogólną w zakresie języków i paradygmatów programowania, programowania obiektowego. Potrafi wybrać paradygmat właściwy dla problemu, który rozwiązuje oraz zna środowiska (języki) umożliwiające implementacje rozwiązania problemu

**EK2 Wiedza** Student zna składnię C, rozumie na czym polega proceduralne podejście do programowania. Ma szczegółową wiedzę nt. algorytmiki, projektowania i programowania obiektowego.

**EK3 Umiejętności** Student potrafi czytać ze zrozumieniem programy napisane w C, potrafi napisać i uruchomić własny program w C, który rozwiązuje postawiony przed nim problem.

**EK4 Umiejętności** Umie stosować styl obiektowy przy tworzeniu prostych programów w języku C przy użyciu struktur i unii, potrafi rozdzielić obsługę tablic, kolejek, stosów od obsługi danych.

## 6 TREŚCI PROGRAMOWE

| WYKŁAD    |                                                                                                                                                                                                                                                                                                  |                  |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| LP        | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH                                                                                                                                                                                                                                           | LICZBA<br>GODZIN |
| <b>W1</b> | Pojęcie języka strukturalnego, struktura programu napisanego w języku C, pliki źródłowe i nagłówkowe. Struktura biblioteki CRT. Pojęcie o IDE. Pojęcie funkcji, funkcji statyczne, strumieni stdin, stdout, znaki specjalne (escape sequence), typy i formaty zmiennych, dyrektywy preprocesora. | 2                |
| <b>W2</b> | Operatory, wyrażenia, instrukcje, operator przypisania. Biblioteka funkcji matematycznych. Konwersja typów, instrukcja warunkowa, operatory arytmetyczne, relacyjne, logiczne, priorytety operatorów.                                                                                            | 2                |
| <b>W3</b> | Definicja a deklaracja funkcji, lista argumentów, argumenty formalne i faktyczne, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcje.                                                                                                                                   | 2                |
| <b>W4</b> | Instrukcje petli for, while, do-while, instrukcja goto. Pojęcie o wskaźnikach. Konwersje typów - jawne i niejawne                                                                                                                                                                                | 2                |
| <b>W5</b> | Otwieranie i zamykanie pliku. Zapis i odczyt sformatowany. Pliki binarne. Sterowanie wskaźnikiem pozycji pliku. Zapis i odczyt do/z pliku binarnego. Tworzenie i usunięcie katalogu. Struktura pliku z rekordami o różnej długości.                                                              | 2                |
| <b>W6</b> | Stałe i zmienne znakowe. Tablice typu char i wchar_t. Stałe tekstowe. Funkcje CRT obsługi tablic. Wskaźniki do tablic. Dynamiczne alokowanie i zwolnienie pamięci dla tablic jednowymiarowych.                                                                                                   | 2                |

| WYKŁAD     |                                                                                                                                                                                                                                                                                            |                  |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| LP         | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH                                                                                                                                                                                                                                     | LICZBA<br>GODZIN |
| <b>W7</b>  | Poruszanie się po tablicy za pomocą wskaźnika. Zmienna enum. Tablice w argumentach funkcji.                                                                                                                                                                                                | 2                |
| <b>W8</b>  | Statyczne wydzielenie pamięci i inicjalizacja. Dynamiczne alokowanie i zwolnienie pamięci. Pojęcie o fragmentowaniu przestrzeni adresowej i sposoby kompresji heap. Kopiowanie tablic na przekrywających się obszarach pamięci oraz inne operacje nad tablicami. Funkcje CRT obsługi heap. | 2                |
| <b>W9</b>  | Tablice o dwóch indeksach. Inicjalizacja tablic. Pojęcie "wskaźnik do wskaźnika". Dynamiczne alokowanie i zwolnienie pamięci dla tablic dwuwymiarowych.                                                                                                                                    | 2                |
| <b>W10</b> | Struktura programu w C. Zmienne lokalne, globalne, statyczne. Klasy pamięci i zakres deklaracji (obszar widoczności).                                                                                                                                                                      | 2                |
| <b>W11</b> | Pojęcie struktury jako łączenie danych w całość. Obiekty struktur oraz tablice struktur. Dynamiczne alokowanie i zwolnienie pamięci.                                                                                                                                                       | 2                |
| <b>W12</b> | Wektory i macierze. Podstawowe algorytmy algebry liniowej i osobliwości ich implementacji w języku C. Rozwijanie petli i blokowanie rejestrów.                                                                                                                                             | 2                |
| <b>W13</b> | Używanie obiektów (danych i funkcji składowych). Projektowanie klas. Inicjalizacja obiektów klas - konstruktory. Przeładowanie konstruktorów                                                                                                                                               | 2                |
| <b>W14</b> | Struktury danych: kolejka, stos, lista. Rozdzielenie obsługi kontenera od obsługi danych. Wskaźniki void * oraz wskaźniki do funkcji. Tablice wskaźników do funkcji.                                                                                                                       | 2                |
| <b>W15</b> | Bitowe operacje logiczne. Maski bitowe.                                                                                                                                                                                                                                                    | 2                |

| LABORATORIUM |                                                                                                                                                  |                  |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| LP           | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH                                                                                           | LICZBA<br>GODZIN |
| <b>L1</b>    | Wprowadzenie danych z klawiatury i wyprowadzenie na monitor. Formatowanie, tworzenie tabeli.                                                     | 2                |
| <b>L2</b>    | Operacje arytmetyczne, priorytet operacji, konwersja typów.                                                                                      | 2                |
| <b>L3</b>    | Obsługa wierszy tekstowych: wprowadzenie danych z monitora, kopiowanie, sklepanie, sortowanie, poszukiwanie kontekstu.                           | 2                |
| <b>L4</b>    | Tablice o dwóch indeksach przy dynamicznym alokowaniu pamięci. Zmiana rozmiaru wiersza, wymiana wierszy pomiędzy sobą, dodawanie nowego wiersza. | 2                |
| <b>L5</b>    | Tablica obiektów struktur przy dynamicznym alokowaniu pamięci. Obsługa tablicy, obsługa obiektu.                                                 | 2                |
| <b>L6</b>    | Wskaźniki do funkcji: program obliczający całkę oznaczoną z gładkiej funkcji jednej zmiennej rzeczywistej.                                       | 2                |

| LABORATORIUM |                                                                                                                                                                                                                                                        |                  |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| LP           | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH                                                                                                                                                                                                 | LICZBA<br>GODZIN |
| <b>L7</b>    | Stworzenie systemu zapisu do pliku binarnego tablicy obiektów. Różne rekordy mają różną długość. Odczyt z pliku binarnego rekordów w dowolnej kolejności.                                                                                              | 2                |
| <b>L8</b>    | Kolokwium I                                                                                                                                                                                                                                            | 2                |
| <b>L9</b>    | Hierarchia obiektów. Obsługa tablicy obiektów Triangle, zawierających obiekt Point (wierzchołek trójkąta). Generalna zasada: kontener (tablica) nie powinna zależeć od typu danych. Technika: stosowanie wskaźników void * oraz wskaźników do funkcji. | 2                |
| <b>L10</b>   | Hierarchia obiektów. Obsługa tablicy obiektów Triangle, zawierających obiekt Point (wierzchołek trójkąta). Teraz Point wymaga dynamicznego alokowania pamięci dla tablicy współrzędnych oraz jej zwolnienia.                                           | 4                |
| <b>L11</b>   | Tworzymy dla poprzedniego zadania menedżer pamięci, odpowiadający za alokowanie, zwolnienie, re-alokowanie w celu przechwytywania ucieczek pamięci.                                                                                                    | 2                |
| <b>L12</b>   | Techniki obsługi macierzy i wektorów. Rozwijanie petli dla algorytmu obliczenia iloczynu skalarnego dwóch wektorów, funkcji pomiaru czasu.                                                                                                             | 2                |
| <b>L13</b>   | Przepisanie jednego obiektu do innego. Powstające problemy i sposoby rozwiązania.                                                                                                                                                                      | 2                |
| <b>L14</b>   | Kolokwium II.                                                                                                                                                                                                                                          | 2                |

## 7 NARZĘDZIA DYDAKTYCZNE

**N1** Wykłady

**N2** Ćwiczenia laboratoryjne

**N3** Konsultacje

**N4** Zadanie projektowe

## 8 OBCIĄŻENIE PRACĄ STUDENTA

| FORMA AKTYWNOŚCI                                                                                 | ŚREDNIA LICZBA GODZIN<br>NA ZREALIZOWANIE<br>AKTYWNOŚCI |
|--------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>Godziny kontaktowe z nauczycielem akademickim, w tym:</b>                                     |                                                         |
| Godziny wynikające z planu studiów                                                               | 60                                                      |
| Konsultacje przedmiotowe                                                                         | 5                                                       |
| Egzaminy i zaliczenia w sesji                                                                    | 0                                                       |
| <b>Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:</b> |                                                         |
| Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury                               | 20                                                      |
| Opracowanie wyników                                                                              | 0                                                       |
| Przygotowanie raportu, projektu, prezentacji, dyskusji                                           | 35                                                      |
| <b>SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z<br/>CAŁEGO NAKŁADU PRACY STUDENTA</b>    | <b>120</b>                                              |
| SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU                                                    | 4.00                                                    |

## 9 SPOSOBY OCENY

### OCENA FORMUJĄCA

**F1** Odpowiedź ustna

**F2** Projekt indywidualny

**F3** Kolokwium

### OCENA PODSUMOWUJĄCA

**P1** Średnia ważona ocen formujących

### WARUNKI ZALICZENIA PRZEDMIOTU

**W1** Zaliczenie projektu indywidualnego

**W2** Napisanie kolokwium na pozytywną ocenę

### KRYTERIA OCENY

| EFEKT KSZTAŁCENIA 1 |                                                                                                                                                                                                                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NA OCENĘ 2.0        | Student nie wie, co to jest paradygmat programowania, nie zna podstawowych paradygmatów, nie zna ważnych języków programowania, nie wie jakie paradygmaty można zrealizować w danym języku, nie potrafi dobrać właściwego paradygmatu i języka do rozwiązania postawionego przed nim zadania |

|                     |                                                                                                                                                                                                                                                                                           |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NA OCENĘ 3.0        | Student zna zasady proceduralnego programowania i potrafi stworzyć prosty program w języku C.                                                                                                                                                                                             |
| NA OCENĘ 3.5        | Student zna zasady proceduralnego programowania i potrafi stworzyć złożony program w języku C, obsłużyć We/Wy do/z pliku binarnego, opracować w programie osobne systemy obsługi błędów i komunikatów.                                                                                    |
| NA OCENĘ 4.0        | Student potrafi - oprócz programowania imperatywnego i obiektowego - programować w logice i zna języki umożliwiające takie podejście do programowania.                                                                                                                                    |
| NA OCENĘ 4.5        | Student zna zasady proceduralnego programowania i potrafi stworzyć program w języku C, używając stylu programowania "obektowego", potrafi stworzyć obsługę dla obiektów o strukturze zagnieżdzonej.                                                                                       |
| NA OCENĘ 5.0        | Student zna zasady proceduralnego programowania i potrafi stworzyć program w języku C, używając stylu programowania "obektowego", potrafi stworzyć obsługę dla obiektów o strukturze zagnieżdzonej, a również rozdzielić funkcjonalności różnych obiektów.                                |
| EFEKT KSZTAŁCENIA 2 |                                                                                                                                                                                                                                                                                           |
| NA OCENĘ 2.0        | Student nie zna wielu podstawowych elementów składni języka C, nie rozumie na czym polega programowanie strukturalne.                                                                                                                                                                     |
| NA OCENĘ 3.0        | Student nie zna wielu elementów składni, ale potrafi je zastąpić innymi; potrafi napisać prosty program w wersji proceduralnej, ma kłopoty ze zrozumieniem działania programów w C na podstawie analizy kodu źródłowego napisanego przez innych programistów                              |
| NA OCENĘ 3.5        | Student opanował składnię C w stopniu pozwalającym na zwarty zapis kodu prostych programów, potrafi analizować kod prostych programów, nie potrafi napisać programu w stylu obiektowym                                                                                                    |
| NA OCENĘ 4.0        | Student potrafi napisać definicję struktury i stworzyć tablicę struktur.                                                                                                                                                                                                                  |
| NA OCENĘ 4.5        | Student zna wszystkie elementy składni C, potrafi zaprojektować program używając styl obiektowy, umie łączyć dane w struktury oraz panuje arytmetyką wskaźników.                                                                                                                          |
| NA OCENĘ 5.0        | Student zna składnię C w stopniu pozwalającym na tworzenie zwartych, szybkich i poprawnie działających programów, potrafi łączyć dane w struktury, dynamicznie alokować i zwalniać pamięć dla tablic obiektów dowolnego typu, umie rozdzielić funkcjonalność kontenera od obsługi danych. |
| EFEKT KSZTAŁCENIA 3 |                                                                                                                                                                                                                                                                                           |
| NA OCENĘ 2.0        | Student nie potrafi analizować kodu źródłowego napisanego przez innych                                                                                                                                                                                                                    |
| NA OCENĘ 3.0        | Student rozumie działanie programu na podstawie lektury kodu źródeł.                                                                                                                                                                                                                      |
| NA OCENĘ 3.5        | Potrafi napisać program w C realizujący przyjęte założenia.                                                                                                                                                                                                                               |
| NA OCENĘ 4.0        | Potrafi zrobić analizę pozwalającą ustalić wymagania jakie stawiamy programowi                                                                                                                                                                                                            |
| NA OCENĘ 4.5        | Potrafi zastosować kod używający dynamicznych struktur danych                                                                                                                                                                                                                             |

|                     |                                                                                                                                                                                                                              |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NA OCENĘ 5.0        | Potrafi napisać kod optymalny pod względem wydajności i uzasadnić jego optymalność                                                                                                                                           |
| EFEKT KSZTAŁCENIA 4 |                                                                                                                                                                                                                              |
| NA OCENĘ 2.0        | Student nie potrafi napisać definicji struktury                                                                                                                                                                              |
| NA OCENĘ 3.0        | Potrafi napisać definicję prostej struktury bez odniesienia się do potrzeb programu                                                                                                                                          |
| NA OCENĘ 3.5        | Student umie zaprojektować strukturę, unie, enumeration dopasowane do potrzeb programu                                                                                                                                       |
| NA OCENĘ 4.0        | Potrafi tworzyć własne biblioteki funkcji                                                                                                                                                                                    |
| NA OCENĘ 4.5        | Potrafi zrobić projekt współdziałania wszystkich struktur i, unij współdziałających w programie, potrafi zamykać dane i funkcje lokalne w pliku i udostępniać działania nad danymi poprzez tworzenie funkcji interfejsowych. |
| NA OCENĘ 5.0        | Student potrafi zaprojektować struktury, unie oraz ich obsługę w stylu obiektowym, które modelują rzeczywiste złożone obiekty, umie definiować struktury, elementami których występują obiekty innych struktur.              |

## 10 MACIERZ REALIZACJI PRZEDMIOTU

| EFEKT KSZTAŁCENIA | ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU | CELE PRZEDMIOTU            | TREŚCI PROGRAMOWE                                                                                | NARZĘDZIA DYDAKTYCZNE | SPOSOBY OCENY |
|-------------------|--------------------------------------------------------------------------------|----------------------------|--------------------------------------------------------------------------------------------------|-----------------------|---------------|
| EK1               | I1_W01<br>I1_W04<br>I1_W06<br>I1_W11                                           | Cel 1 Cel 2<br>Cel 3 Cel 4 | W1 W2 W11 L1<br>L2 L3 L4 L5 L6<br>L7 L8 L9 L10<br>L11 L12 L13 L14                                | N1 N3                 | F2            |
| EK2               | I1_W04<br>I1_W06                                                               | Cel 2 Cel 3<br>Cel 4       | W1 W2 W3 W4<br>W5 W6 W7 W8<br>W9 W10 W11<br>L1 L2 L3 L4 L5<br>L6 L7 L8 L9 L10<br>L11 L12 L13 L14 | N1 N2 N3              | F1 F3         |
| EK3               | I1_U01b<br>I1_U04<br>I1_U06b<br>I1_U10                                         | Cel 2                      | W2 W3 W4 W5<br>W6 W7 W8 W9<br>W10 W11 L1 L2<br>L3 L4 L5 L6 L7<br>L8 L9 L10 L11<br>L12 L13 L14    | N1 N2 N3              | F1 F3         |

| EFEKT KSZTAŁCENIA | ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU | CELE PRZEDMIOTU | TREŚCI PROGRAMOWE                            | NARZĘDZIA DYDAKTYCZNE | SPOSOBY OCENY |
|-------------------|--------------------------------------------------------------------------------|-----------------|----------------------------------------------|-----------------------|---------------|
| EK4               | I1_U01b<br>I1_U04<br>I1_U06b<br>I1_U07b                                        | Cel 3 Cel 4     | W1 W2 W3 W11<br>L8 L9 L10 L11<br>L12 L13 L14 | N1 N2 N3              | F1            |

## 11 WYKAZ LITERATURY

### LITERATURA PODSTAWOWA

- [1] | B.Kernighan, D.Ritchie — *Język ANSI C*, Warszawa, 2002, WNT
- [2] | J. Richter, C. Nasarre. — *Windows via C/C++*, Microsoft Press, 2008, APN PROMISE

### LITERATURA UZUPEŁNIAJĄCA

- [1] | S.Prata — *Język C. Szkoła Programowania*, Gliwice, 2006, Helion
- [2] | — *MSDN - Microsoft Developer Network*, , 2020, Dokumentacja Microsoft

## 12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

### OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr hab. inż. prof.PK. Sergiej Fialko (kontakt: sfialko@pk.edu.pl)

### OSOBY PROWADZĄCE PRZEDMIOT

- 1 dr hab. inż. Siergiy Fialko (kontakt: sfialko@pk.edu.pl)
- 2 mgr. inż Jan Wojtas (kontakt: jwojtas@pk.edu.pl)
- 3 mgr. Marcin Wątopek (kontakt: marcin.watopek@pk.edu.pl)
- 4 mgr. inż Łukasz Gaża (kontakt: lukasz.gaza@pk.edu.pl)

## 13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)





**PRZYJMUJĘ DO REALIZACJI** (data i podpisy osób prowadzących przedmiot)

.....  
.....  
.....  
.....