

# POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

## KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2014/2015

Wydział Fizyki, Matematyki i Informatyki

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: niestacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

### 1 INFORMACJE O PRZEDMIOCIE

|                                         |                                      |
|-----------------------------------------|--------------------------------------|
| NAZWA PRZEDMIOTU                        | Języki i paradygmaty programowania 1 |
| NAZWA PRZEDMIOTU<br>W JĘZYKU ANGIELSKIM |                                      |
| KOD PRZEDMIOTU                          | WFMiI I oIN C4 14/15                 |
| 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       | 18     | 0         | 18           | 0                                | 0          | 0       |

### 3 CELE PRZEDMIOTU

**Cel 1** Poznanie podstawowych paradygmatów programowania, w tym paradygmatu imperatywnego (proceduralnego) oraz obiektowego, funkcyjnego i programowania w logice

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

**Cel 3** Podstawy obiektowego podejścia do programowania - abstrakcja danych, enkapsulacja, ukrywanie implementacji. Rozszerzenia w C++ pozwalające na programowanie obiektowe.

**Cel 4** Projektowanie klas, tworzenie programów obiektowych

## 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, zna środowiska (języki) umożliwiające ich implementację, potrafi wybrać paradygmat właściwy dla problemu, który rozwiązuje.

**EK2 Wiedza** Student zna składnię C i C++, rozumie na czym polega obiektowe podejście do programowania

**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** Student potrafi zaprojektować klasę modelującą rzeczywiste obiekty i umie stosować w programie klasy napisane przez siebie oraz klasy biblioteczne.

## 6 TREŚCI PROGRAMOWE

| WYKŁAD    |                                                                                                                                                                                                                    |                  |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| LP        | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH                                                                                                                                                             | LICZBA<br>GODZIN |
| <b>W1</b> | Pojęcie paradygmat programowania, przegląd najważniejszych paradygmatów. Formalna definicja języka programowania. Przegląd języków i paradygmatów, które można w nich realizować                                   | 2                |
| <b>W2</b> | Zasady przetwarzania programu w C/C++. Instrukcje we/wy, słowa kluczowe języka, Przykłady prostych programów. Typy w C. Problem unikalności nazw - pojęcie przestrzeni nazw.                                       | 2                |
| <b>W3</b> | Operatory w C; sposób działania, ilość argumentów, priorytet, typ łączności. Specyfikatory dostępu do pamięci. Konwersje typów - jawne i niejawne                                                                  | 1                |
| <b>W4</b> | Funkcje i struktura programu. Definicja i wywołanie funkcji, przekazywanie argumentów. Zmienne globalne i lokalne. Sposób podziału kodu źródłowego na pliki. Domyślne argumenty funkcji, funkcje inline. Rekursja. | 2                |
| <b>W5</b> | Tablice definicje, inicjalizacja, dostęp do składowych. Tablice jak argumenty funkcji. Tablice wielowymiarowe, tablice znakowe, C-stringi.                                                                         | 1                |
| <b>W6</b> | Wskaźniki i ich zastosowanie do przekazywania argumentów funkcji, usprawnienia pracy z tablicami, dynamicznej alokacji pamięci, uzyskania dostępu do wybranych komórek pamięci.                                    | 2                |
| <b>W7</b> | 8. Dynamiczna alokacja tablic dwuwymiarowych. Wskaźniki do funkcji. Zasady tworzenia i interpretacji deklaracji złożonych. Przekazywanie argumentów programu (funkcji main() ) z wiersza poleceń                   | 2                |

| WYKŁAD     |                                                                                                                                                 |                  |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| LP         | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH                                                                                          | LICZBA<br>GODZIN |
| <b>W8</b>  | Przeładowanie (przeciążenie) nazw funkcji (2 godz). Praca z plikami zewnętrznymi                                                                | 1                |
| <b>W9</b>  | Problemy programowania proceduralnego i ich źródła. Obiekt - abstrakcja rzeczywistego obiektu. Klasa - nowy typ danych - opis stanu i zachowań. | 2                |
| <b>W10</b> | Definicja klasy. Pojęcie danych i funkcji składowych. Enkapsulacja, etykiety dostępu. Sposoby definiowania funkcji składowych                   | 2                |
| <b>W11</b> | Używanie obiektów (danych i funkcji składowych). Projektowanie klas.                                                                            | 1                |

| LABORATORIUM |                                                                                                                                                                                                                                                                                                     |                  |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| LP           | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH                                                                                                                                                                                                                                              | LICZBA<br>GODZIN |
| <b>L1</b>    | Powtórzenie wiadomości o podstawowych instrukcjach sterujących, instrukcjach we/wy oraz operatorach w języku C. Proste programy wyrabiające biegłość w stosowaniu instrukcji formatowanego we/wy, pętli i instrukcji warunkowych                                                                    | 2                |
| <b>L2</b>    | Programy przetwarzające znaki i łańcuchy znaków (zliczanie ilości linii, słów, znaków, białych znaków, cyfr, liter). Odwracanie łańcucha znaków. konwersja liczba całkowita - ciąg znaków                                                                                                           | 2                |
| <b>L3</b>    | Operatory bitowe. Napisanie funkcji, które: wyświetlają obraz bitowy słowa, ustawiają, kasują lub negują grupę bitów, realizują 32-bitowy bufor cykliczny z przesunięciem w lewo                                                                                                                    | 2                |
| <b>L4</b>    | Tablice dwuwymiarowe; definicja statyczna i dynamiczna program wczytujący tablicę (macierz) 4 x 4 liczb i obliczający wyznacznik, wykorzystujący statyczną i dynamiczną deklarację tablic.                                                                                                          | 2                |
| <b>L5</b>    | Wskaźniki do funkcji: program obliczający całkę oznaczoną z gładkiej funkcji jednej zmiennej rzeczywistej.                                                                                                                                                                                          | 2                |
| <b>L6</b>    | Rekursja, wskaźniki, dynamiczna alokacja pamięci: program wykorzystujący listę jednokierunkową nazwisk, pozwalający na sortowane, drukowanie wskazanego kawałka listy, wybranie elementu spełniającego postawione wymagania                                                                         | 2                |
| <b>L7</b>    | Klasy w C++ : zaprojektowanie klasy opisującej dane studenta. Klasa ma być wyposażona w metody pozwalające na sortowanie wg. dowolnego pola, drukowanie listy, wyznaczenie najmłodszego/najstarszego studenta w grupie. Dynamicznie utworzone obiekty tej klasy zestawiamy w listę jednokierunkową. | 2                |
| <b>L8</b>    | Struktury drzewiaste: należy zaprojektować klasę do opisu drzewa genealogicznego. Metody klasy mają pozwolić na: dodawanie osoby do drzewa, edycję danych, zapis/odczyt do/z pliku tekstowego.                                                                                                      | 2                |
| <b>L9</b>    | Kollokwium 1                                                                                                                                                                                                                                                                                        | 2                |

## 7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

## 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                                                               | 36                                                      |
| Konsultacje przedmiotowe                                                                         | 20                                                      |
| Egzaminy i zaliczenia w sesji                                                                    | 10                                                      |
| <b>Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:</b> |                                                         |
| Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury                               | 30                                                      |
| Opracowanie wyników                                                                              | 0                                                       |
| Przygotowanie raportu, projektu, prezentacji, dyskusji                                           | 20                                                      |
| <b>SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z<br/>CAŁEGO NAKŁADU PRACY STUDENTA</b>    | <b>116</b>                                              |
| SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU                                                    | 4.00                                                    |

## 9 SPOSOBY OCENY

### OCENA FORMUJĄCA

F1 Sprawozdanie z ćwiczenia laboratoryjnego

F2 Odpowiedź ustna

F3 Projekt indywidualny

F4 Kolokwium

### OCENA PODSUMOWUJĄCA

P1 Średnia ważona ocen formujących

### WARUNKI ZALICZENIA PRZEDMIOTU

W1 Zaliczenie wszystkich projektów indywidualnych

### 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 paradygmat imperatywny programowania i potrafi go zaimplementować w proceduralnym języku strukturalnym                                                                                                                                                                           |
| NA OCENĘ 3.5        | Student - oprócz paradygmatu imperatywnego - zna obiektowy paradygmat programowania, wie jakich języków powinien użyć do wykonania implementacji                                                                                                                                             |
| 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 i potrafi zastosować wszystkie podstawowe paradygmaty programowania                                                                                                                                                                                                              |
| NA OCENĘ 5.0        | Student zna i potrafi zastosować wszystkie podstawowe paradygmaty programowania, potrafi dokonać analizy i wybrać właściwy paradygmat i język programowania do jego implementacji oraz potrafi swój wybór przekonywująco uzasadnić/                                                          |
| EFEKT KSZTAŁCENIA 2 |                                                                                                                                                                                                                                                                                              |
| NA OCENĘ 2.0        | Student nie zna wielu podstawowych elementów składni języka C/C++, nie rozumie na czym polega programowanie obiektowe                                                                                                                                                                        |
| 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/C++ na podstawie analizy kodu źródłowego napisanego przez innych programistów                             |
| NA OCENĘ 3.5        | Student opanował składnię C/C++ w stopniu pozwalającym na zwarty zapis kodu prostych programów, potrafi analizować kod prostych programów, nie potrafi napisać programu obiektowego                                                                                                          |
| NA OCENĘ 4.0        | Student potrafi napisać definicję prostej klasy będącej zawierającej podstawowe elementy opisu i wykonującego podstawowe działania na danych                                                                                                                                                 |
| NA OCENĘ 4.5        | Student zna wszystkie elementy składni C/C++ , potrafi zaprojektować klasy potrzebne w implementacji programu w wersji obiektowej,                                                                                                                                                           |
| NA OCENĘ 5.0        | Student zna składnię C/C++ w stopniu pozwalającym na tworzenie zwartych, szybkich i poprawnie działających programów, potrafi zaprojektować i dokonać implementacji metod klas w podejściu obiektowym.                                                                                       |
| 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 klasy                                                                                                         |
| NA OCENĘ 3.0        | Potrafi napisać definicję prostej klasy bez odniesienia się do potrzeb programu                                                                     |
| NA OCENĘ 3.5        | Student umie zaprojektować klasę dopasowaną do potrzeb programu                                                                                     |
| NA OCENĘ 4.0        | Potrafi wykorzystać klasy biblioteczne                                                                                                              |
| NA OCENĘ 4.5        | Potrafi zrobić projekt współdziałania wszystkich klas współdziałających w programie                                                                 |
| NA OCENĘ 5.0        | Student potrafi zaprojektować klasę modelującą rzeczywiste obiekty i umie stosować w programie klasy napisane przez siebie oraz klasy biblioteczne. |

## 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_W06,<br>I1_W08,<br>I1_W10,<br>I1_W13,<br>I1_U07,<br>I1_U08,<br>I1_U15, I1_U20                       | Cel 1 Cel 2<br>Cel 3 Cel 4 | W1 W11                                   | N1 N3                 | F2 F4         |
| EK2               | I1_W06,<br>I1_W08,<br>I1_W10,<br>I1_W13,<br>I1_U07,<br>I1_U12,<br>I1_U16,<br>I1_U21,<br>I1_U23, I1_K02 | Cel 2 Cel 3<br>Cel 4       | W1 W2 W3 W4<br>W5 W6 W7 W8<br>W9 W10 W11 | N1 N2 N3              | F1 F3 F4      |

| EFEKT KSZTAŁCENIA | ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU | CELE PRZEDMIOTU | TREŚCI PROGRAMOWE                                    | NARZĘDZIA DYDAKTYCZNE | SPOSOBY OCENY |
|-------------------|--------------------------------------------------------------------------------|-----------------|------------------------------------------------------|-----------------------|---------------|
| EK3               | I1_W06,<br>I1_W10,<br>I1_U16                                                   | Cel 2           | W2 W3 W4 W5<br>W6 W7 W8 W9<br>W10 W11 L1 L2<br>L3 L4 | N1 N2 N3              | F1 F3 F4      |
| EK4               | I1_W06,<br>I1_W10,<br>I1_U08, I1_U12                                           | Cel 3 Cel 4     | W1 W11 L8 L9                                         | N1 N2 N3              | F1 F4         |

## 11 WYKAZ LITERATURY

### LITERATURA PODSTAWOWA

- [1] | B.Kernighan, D.Ritchie — *Język ANSI C*, Warszawa, 2002, WNT  
[2] | J.Grębosz — *Symfonia C++ Standard*, Kraków, 2006, Editions 2000

### LITERATURA UZUPEŁNIAJĄCA

- [1] | B.Eckel — *Thinking in C++*, Gliwice, 2002, Helion  
[2] | S.Prata — *Język C. Szkoła Programowania*, Gliwice, 2006, Helion

## 12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

### OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr inż. Piotr Poznański (kontakt: [poznan@pk.edu.pl](mailto:poznan@pk.edu.pl))

### OSOBY PROWADZĄCE PRZEDMIOT

- 1 dr hab. Janusz Chwastowski (kontakt: [jchwastowski@pk.edu.pl](mailto:jchwastowski@pk.edu.pl))  
2 mgr inż. Marek Kosiorowski (kontakt: )

## 13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....  
.....