

# POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

## KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2013/2014

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 2 |
| NAZWA PRZEDMIOTU<br>W JĘZYKU ANGIELSKIM |                                      |
| KOD PRZEDMIOTU                          | WFMiI I oIN C4 13/14                 |
| KATEGORIA PRZEDMIOTU                    | Przedmioty kierunkowe                |
| LICZBA PUNKTÓW ECTS                     | 6.00                                 |
| SEMESTRY                                | 3                                    |

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

| SEMESTR | WYKŁAD | ĆWICZENIA | LABORATORIUM | LABORATORIUM<br>KOMPUTERO-<br>WE | SEMINARIUM | PROJEKT |
|---------|--------|-----------|--------------|----------------------------------|------------|---------|
| 3       | 18     | 0         | 18           | 0                                | 0          | 0       |

### 3 CELE PRZEDMIOTU

**Cel 1** Osiągnięcia umiejętności w ocenie przydatności paradygmatów i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów; projektowania, implementacji, testowania i debugowania prostych programów obiektowych.

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

1 Język C, podstawowa wiedza z analizy matematycznej, algebry liniowej

## 5 EFEKTY KSZTAŁCENIA

**EK1 Wiedza** Ma wiedzę ogólną w zakresie języków i paradygmatów programowania, programowania obiektowego

**EK2 Wiedza** Ma szczegółową wiedzę nt. algorytmiki, projektowania i programowania obiektowego.

**EK3 Umiejętności** Umie stworzyć model obiektowy prostych programów w języku C++ przy użyciu klas, dziedziczenia, wielodziedziczenia, przeciążenia funkcji i operatorów, szablonów funkcji i klas, identyfikacji typów, wyjątków.

**EK4 Umiejętności** Ma umiejętność formułowania algorytmów i ich programowania. Wykazuje umiejętności tworzenia i debugowania programów w środowisku IDE Microsoft Visual Studio 2010.

## 6 TREŚCI PROGRAMOWE

| WYKŁAD    |  |                  |
|-----------|--|------------------|
| LP        | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH   | LICZBA<br>GODZIN |
| <b>W1</b> | Oprogramowanie obiektowe: inkapsulacja, polimorfizm i dziedziczenie. 2. Zasięg deklaracji i czas trwania obiektów. Przestrzeń nazw. Wprowadzenie do klas. Konstruktorzy i destruktory. | 2                |
| <b>W2</b> | Wprowadzenie w dziedziczenie. Specyfikatory private, protected, public. Funkcje inline. Przypisanie obiektów. Przekazywanie obiektów do funkcji.                                       | 2                |
| <b>W3</b> | Zwracanie obiektu przez funkcje. Funkcje zaprzyjaźnione. Przeciążenie funkcji, konstruktorzy kopii, argumenty domyślne.  | 2                |
| <b>W4</b> | Przeciążenie operatorów.   | 2                |
| <b>W5</b> | Dynamiczne alokowanie pamięci. Operatory new, delete. Dziedziczenie. Specyfikatory dostępu. Wielodziedziczenie. Klasy wirtualne.   | 2                |
| <b>W6</b> | Polimorfizm dynamiczny. Funkcje wirtualne, abstrakcyjne, klasy abstrakcyjne.   | 2                |
| <b>W7</b> | Identyfikacje typu na etapie wykonania (RTTI) .  | 2                |
| <b>W8</b> | Szablony funkcji i klas.   | 2                |
| <b>W9</b> | Wejście-wyjście w C++. Obsługa wyjątków. Statyczne składowe klasy. Specyfikatory const, volatile. Niepolimorficzne rzutowanie typów.   | 2                |

| LABORATORIUM |  |                  |
|--------------|--|------------------|
| LP           | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA<br>GODZIN |

| LABORATORIUM |   |                  |
|--------------|---|------------------|
| LP           | TEMATYKA ZAJĘĆ<br>OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH  | LICZBA<br>GODZIN |
| L1           | Wprowadzenie w C++. Pierwsze programy. Inkapsulacja. Pojęcia klasy. Dane i metody klasy   | 2                |
| L2           | Inicjowanie i niszczenie obiektu. Konstruktorzy i destruktory. Hierarchia klas, wprowadzenie w dziedziczenie.                                 | 2                |
| L3           | Funkcje inline. Funkcje zaprzyjaźnione do klasy   | 2                |
| L4           | Operatory new, delete. Alokowanie i zwolnienie pamięci.   | 2                |
| L5           | Funkcje zaprzyjaźnione do klasy. Przeciążenie operatorów binarnych.   | 2                |
| L6           | Operatory new, delete. Alokowanie i zwolnienie pamięci.   | 2                |
| L7           | Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez referencje). Konstruktorzy kopii. Przeciążenie operatora przypisania. | 2                |
| L8           | Funkcje-szablony.   | 2                |
| L9           | Funkcje wirtualne, identyfikacje typu RTTI.   | 2                |

## 7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

N4 Prezentacje multimedialne

N5 Inne

## 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   | 0   |
| Konsultacje przedmiotowe   | 18  |
| 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                               | 46  |
| Opracowanie wyników  | 20  |
| Przygotowanie raportu, projektu, prezentacji, dyskusji   | 50  |
| <b>SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA</b>        | <b>144</b>  |
| SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU  | 6.00  |

## 9 SPOSOBY OCENY

### OCENA FORMUJĄCA

F1 Ćwiczenie praktyczne

F2 Sprawozdanie z ćwiczenia laboratoryjnego

F3 Zadanie tablicowe

F4 Odpowiedź ustna

### OCENA PODSUMOWUJĄCA

P1 Egzamin praktyczny

### OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA

B1 Projekt indywidualny

### KRYTERIA OCENY

| EFEKT KSZTAŁCENIA 1 |   |
|---------------------|---|
| NA OCENĘ 2.0        | Student nie zna podstawowych pojęć z zakresu albo inkapsulacji, albo dziedziczenia albo polimorfizmu. |

|                     |  |
|---------------------|--|
| NA OCENĘ 3.0        | Student zna podstawowe pojęcia z zakresu inkapsulacji, dziedziczenia i polimorfizmu.   |
| NA OCENĘ 3.5        | Student zna podstawowe pojęcia z zakresu inkapsulacji, dziedziczenia i polimorfizmu, wykazuje znajomość tworzenia prostych klas C++ oraz przeciążenia funkcji i operatorów.  |
| NA OCENĘ 4.0        | Student zna podstawowe pojęcia z zakresu inkapsulacji, dziedziczenia i polimorfizmu, wykazuje znajomość tworzenia klasy C++, zawierającej wskaźniki do obiektów prostych typów, przeciążenia funkcji i operatorów, tworzenia klas pochodnych na podstawie wielodziedziczenia.  |
| NA OCENĘ 4.5        | Student zna podstawowe pojęcia z zakresu inkapsulacji, dziedziczenia i polimorfizmu, wykazuje znajomość tworzenia klasy C++, zawierającej wskaźniki do obiektów prostych typów, przeciążenia funkcji i operatory, tworzenia klas pochodnych na podstawie wielodziedziczenia, funkcji wirtualnych oraz klas abstrakcyjnych.   |
| NA OCENĘ 5.0        | Student zna podstawowe pojęcia z zakresu inkapsulacji, dziedziczenia i polimorfizmu, wykazuje znajomość tworzenia klasy C++, zawierającej wskaźniki do obiektów prostych typów oraz typów innych klas i obiektów STL, przeciążenia funkcji i operatory, tworzenia klas pochodnych na podstawie wielodziedziczenia, funkcji wirtualnych oraz klas abstrakcyjnych, potrafi przekonująco wytłumaczyć, dla czego w podanym przypadku trzeba postępować tak, a nie inaczej. |
| EFEKT KSZTAŁCENIA 2 |  |
| NA OCENĘ 2.0        | Student nie zna albo podstawowych pojęć z zakresu szablonów funkcji i klas, albo formatowanego We/Wy.  |
| NA OCENĘ 3.0        | Student zna podstawowe pojęcia z zakresu szablonów funkcji i klas i formatowanego We/Wy.   |
| NA OCENĘ 3.5        | Student zna podstawowe pojęcia z zakresu szablonów funkcji i klas, formatowanego We/Wy, wykazuje podstawowe znajomości z klasami-kontenerami STL oraz z algorytmami STL.   |
| NA OCENĘ 4.0        | Student zna podstawowe pojęcia z zakresu szablonów funkcji i klas, zidentyfikowania typów w czasie wykonania programu, formatowanego oraz nieformatowanego We/Wy, wykazuje podstawowe znajomości z klasami-kontenerami STL oraz z algorytmami STL.   |
| NA OCENĘ 4.5        | Student zna podstawowe pojęcia z zakresu szablonów funkcji i klas, zidentyfikowania typów w czasie wykonania programu, formatowanego oraz nieformatowanego We/Wy, wykazuje podstawowe znajomości z klasami-kontenerami STL oraz z algorytmami STL, wyjątkami.  |
| NA OCENĘ 5.0        | Student doskonale zna podstawowe pojęcia z zakresu szablonów funkcji i klas, zidentyfikowania typów w czasie wykonania programu, formatowanego oraz nieformatowanego We/Wy, wykazuje podstawowe znajomości z klasami-kontenerami STL oraz z algorytmami STL, wyjątkami, potrafi przekonująco wytłumaczyć, dla czego w podanym przypadku trzeba postępować tak, a nie inaczej.  |
| EFEKT KSZTAŁCENIA 3 |  |

|                     |   |
|---------------------|---|
| NA OCENĘ 2.0        | Student nie umie tworzyć prostych programów przy użyciu klas, prostego dziedziczenia, przeciążenia funkcji i operatorów.  |
| NA OCENĘ 3.0        | Student umie tworzyć proste programy przy użyciu klas, prostego dziedziczenia, przeciążenia funkcji.  |
| NA OCENĘ 3.5        | Student umie tworzyć proste programy przy użyciu klas, prostego dziedziczenia, przeciążenia funkcji i operatorów, przekazywać obiekty do funkcji przez wartość, przez wskaźnik i przez referencje.  |
| NA OCENĘ 4.0        | Student umie tworzyć proste programy przy użyciu klas oraz klas, zawierających wskaźniki do obiektów prostych typów, używać wielodziedziczenie, przeciążenie funkcji i operatorów, tworzyć konstruktory kopiujące, przekazywać obiekty do funkcji przez wartość, przez wskaźnik i przez referencje.   |
| NA OCENĘ 4.5        | Student umie tworzyć proste programy, składające się z kilku plików, przy użyciu klas prostych oraz klas, zawierających wskaźniki do obiektów prostych typów, używać wielodziedziczenie, przeciążenie funkcji i operatorów, tworzyć konstruktory kopiujące, przekazywać obiekty do funkcji przez wartość, przez wskaźnik i przez referencje, tworzyć funkcje wirtualne, destruktory wirtualne, klasy abstrakcyjne, stosować polimorfizm dynamiczny.   |
| NA OCENĘ 5.0        | Student umie tworzyć proste programy, składające się z kilku plików, przy użyciu klas prostych oraz klas, zawierających wskaźniki do obiektów prostych typów oraz typów innych klas i obiektów STL, używać wielodziedziczenie, przeciążenie funkcji i operatorów, tworzyć konstruktory kopiujące, przekazywać obiekty do funkcji przez wartość, przez wskaźnik i przez referencje, tworzyć funkcje wirtualne, destruktory wirtualne, klasy abstrakcyjne, stosować polimorfizm dynamiczny, potrafi przekonująco wytłumaczyć, dla czego w podanym przypadku trzeba postępować tak, a nie inaczej. |
| EFEKT KSZTAŁCENIA 4 |   |
| NA OCENĘ 2.0        | Student nie umie tworzyć szablonów funkcji i klas albo nie umie stosować formatowane We/Wy.   |
| NA OCENĘ 3.0        | Student umie tworzyć szablonów funkcji i klas, stosować formatowane We/Wy.  |
| NA OCENĘ 3.5        | Student umie tworzyć szablonów funkcji i klas, stosować formatowane oraz nieformatowane We/Wy, używać klasy-kontenery oraz algorytmy STL do obiektów prostych typów.  |
| NA OCENĘ 4.0        | Student umie tworzyć szablonów funkcji i klas, stosować formatowane We/Wy, używać klasy-kontenery oraz algorytmy STL do obiektów prostych typów, umie używać operatory zidentyfikowania typów RTTI, stosować obiekty klas polimorficznych.  |
| NA OCENĘ 4.5        | Student umie tworzyć szablonów funkcji i klas, stosować formatowane We/Wy, używać klasy-kontenery oraz algorytmy STL do obiektów klas, umie używać operatory zidentyfikowania typów RTTI, stosować obiekty klas polimorficznych, stosować wyjątki.  |

|              |   |
|--------------|---|
| NA OCENĘ 5.0 | Student umie tworzyć szablony funkcji i klas, stosować formatowane We/Wy, używać klasy-kontenery oraz algorytmy STL do obiektów klas, umie używać operatory zidentyfikowania typów RTTI, stosować obiekty klas polimorficznych, stosować wyjątki, potrafi przekonująco wytłumaczyć, dla czego w podanym przypadku trzeba postępować tak, a nie inaczej. |
|--------------|---|

## 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   | Cel 1           | W1 W2 W3 W4 W5    | N1 N2 N3 N4           | F1 F2 F4 P1    |
| EK2               | I1_W08   | Cel 1           | W6 W7 W8 W9       | N1 N2 N3 N4 N5        | F1 F2 F3 F4 P1 |
| EK3               | I1_U08   | Cel 1           | L1 L2 L3 L4 L5    | N1 N2 N3 N5           | F1 F2 F4 P1    |
| EK4               | I1_U07   | Cel 1           | L6 L7 L8 L9       | N1 N2 N3 N5           | F1 F2 F3 F4 P1 |

## 11 WYKAZ LITERATURY

### LITERATURA PODSTAWOWA

- [1 ] **B. Stroustrup** — *Język C++*, Warszawa, 2000, WNT
- [2 ] **H. Schildt** — *Programowanie C++*, Warszawa,, 2002, WNT
- [3 ] **Microsoft Corp** — *MSDN*, -, 2011, -

### LITERATURA UZUPEŁNIAJĄCA

- [1 ] **S. B. Lippman, J. Lajoie** — *Podstawy języka C++*, Warszawa, 2003, WNT

## 12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

### OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr hab. inż. Sergiy Fialko (kontakt: [sfialko@pk.edu.pl](mailto:sfialko@pk.edu.pl))

## OSOBY PROWADZĄCE PRZEDMIOT

1 dr. hab. inż. Sergiy Fialko (kontakt: [sfialko@riad.pk.edu.pl](mailto:sfialko@riad.pk.edu.pl))

## 13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

---

(miejsowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....