

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2012/2013

Wydział Fizyki, Matematyki i Informatyki

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 2 |
| NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM | |
| KOD PRZEDMIOTU | WFMiI I oIS C4 12/13 |
| KATEGORIA PRZEDMIOTU | Przedmioty kierunkowe |
| LICZBA PUNKTÓW ECTS | 6.00 |
| SEMESTRY | 3 |

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

| SEMESTR | WYKŁAD | ĆWICZENIA | LABORATORIUM | LABORATORIUM KOMPUTERO- WE | SEMINARIUM | PROJEKT |
|---------|--------|-----------|--------------|----------------------------------|------------|---------|
| 3 | 30 | 0 | 30 | 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 OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| W1 | Oprogramowanie obiektowe: inkapsulacja, polimorfizm i dziedziczenie. 2. Zasięg deklaracji i czas trwania obiektów. Przestrzeń nazw. | 2 |
| W2 | Wprowadzenie do klas. Konstruktorzy i destruktory. | 2 |
| W3 | Wprowadzenie w dziedziczenie. Specyfikatory private, protected, public. Funkcje inline. Przypisanie obiektów. Przekazywanie obiektów do funkcji. | 2 |
| W4 | Zwracanie obiektu przez funkcje. Funkcje zaprzyjaźnione. | 2 |
| W5 | Przeciążenie funkcji, konstruktorzy kopii, argumenty domyślne. | 2 |
| W6 | Przeciążenie operatorów. | 2 |
| W7 | Dynamiczne alokowanie pamięci. Operatory new, delete. | 2 |
| W8 | Dziedziczenie. Specyfikatory dostępu. Wielodziedziczenie. Klasy wirtualne. | 2 |
| W9 | Polimorfizm dynamiczny. Funkcje wirtualne, abstrakcyjne, klasy abstrakcyjne. | 2 |
| W10 | Szablony funkcji i klas. | 2 |
| W11 | Identyfikacje typu na etapie wykonania (RTTI) . | 3 |
| W12 | Wejście-wyjście w C++. | 3 |
| W13 | Obsługa wyjątków. Statyczne składowe klasy. Specyfikatory const, volatile. Niepolimorficzne rzutowanie typów. | 2 |
| W14 | Wprowadzenie w STL | 2 |

| LABORATORIUM | | |
|--------------|--|------------------|
| LP | TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| L1 | Wprowadzenie w C++. Pierwsze programy. | 2 |
| L2 | Inkapsulacja. Pojęcia klasy. Dane i metody klasy | 2 |
| L3 | Inicjowanie i niszczenie obiektu. Konstruktorzy i destruktory. | 2 |
| L4 | Hierarchia klas, wprowadzenie w dziedziczenie. Funkcji inline. | 2 |
| L5 | Funkcji zaprzyjaźnione do klasy | 2 |
| L6 | Operatory new, delete. Alokowanie i zwolnienie pamięci. | 2 |
| L7 | Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez referencje). Konstruktory kopii. | 2 |
| L8 | Przeciążenie operatorów binarnych. Przeciążenie operatora przypisania. | 2 |
| L9 | Funkcji-szablony. | 2 |
| L10 | Klasa-szablon my_vect. Tworzenie projektu z wieloma plikami. Klasy obsługi komunikatów i interfejsu. Klasy danych. | 2 |
| L11 | Praca z plikami tekstowymi i binarnymi. Odczyt i zapis dokumentu. | 2 |
| L12 | Funkcje wirtualne, identyfikacje typu RTTI. | 4 |
| L13 | Kolokwium 1 | 2 |
| L14 | kolokwium 2 | 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 NA ZREALIZOWANIE AKTYWNOŚCI |
|--|---|
| Godziny kontaktowe z nauczycielem akademickim, w tym: | |
| Godziny wynikające z planu studiów | 0 |
| Konsultacje przedmiotowe | 18 |
| Egzaminy i zaliczenia w sesji | 10 |
| Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym: | |
| Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury | 30 |
| Opracowanie wyników | 12 |
| Przygotowanie raportu, projektu, prezentacji, dyskusji | 50 |
| SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA | 120 |
| SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU | 6.00 |

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Kolokwium

F2 Zadanie tablicowe

F3 Projekt indywidualny

F4 Test

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 SZCZEGÓLOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU | CELE PRZEDMIOTU | TREŚCI PROGRAMOWE | NARZĘDZIA DYDAKTYCZNE | SPOSOBY OCENY |
|-------------------|--|-----------------|------------------------------------|-----------------------|----------------|
| EK1 | I1_W06 | Cel 1 | W1 W2 W3 W4 W5 W6 W7 | N1 N2 N3 N4 | F1 F2 F4 P1 |
| EK2 | I1_W08 | Cel 1 | W7 W8 W9 W10 W11 W12 W13 W14 | N1 N2 N3 N4 N5 | F1 F2 F3 F4 P1 |
| EK3 | I1_U08 | Cel 1 | L1 L2 L3 L4 L5 L6 L7 L8 | N1 N2 N3 N5 | F1 F2 F4 P1 |
| EK4 | I1_U07 | Cel 1 | L9 L10 L11 L12 L13 L14 | 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. Sergij Fialko (kontakt: sfialko@pk.edu.pl)



OSOBY PROWADZĄCE PRZEDMIOT

1 dr. hab. inż. Sergiy Fialko (kontakt: 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)

.....