

# PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA W NOWYM SĄCZU

## KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2015/2016

Instytut Techniczny

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: Stacjonarne

Kod kierunku: 11.3

Stopień studiów: I

Specjalności: Informatyka stosowana

### 1 PRZEDMIOT

NAZWA PRZEDMIOTU	Zaawansowane programowanie
KOD PRZEDMIOTU	IT 11.3 AIS B3 15/16
KATEGORIA PRZEDMIOTU	Przedmioty podstawowe i kierunkowe
LICZBA PUNKTÓW ECTS	5
SEMESTRY	4

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

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	PROJEKT	SEMINARIUM
4	15			30	

### 3 CELE PRZEDMIOTU

**Cel 1** Opanowanie zaawansowanych i nowoczesnych metod programowania (w oparciu o język C++). Stosowanie wzorców projektowych jako optymalnych rozwiązań problemu.

**Cel 2** Posługiwanie się iteratorami i kontenerami biblioteki STL. Alokowanie pamięci, wykorzystanie bezpiecznych i inteligentnych wskaźników.

**Cel 3** Programowanie funktorami (obiekty funkcyjne, wyrażenia lambda). Wykorzystanie algorytmów biblioteki standardowej.

**Cel 4** Programowanie szablonami. Umiejętność testowania i śledzenia oprogramowania.

**Cel 5** Programowanie wielowątkowe (w standardzie C++14) i zarządzanie zasobami współdzielonymi.



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

- a Ugruntowana znajomość podstaw języka C++ oraz biblioteki standardowej. Znajomość technik i paradygmatów programowania obiektowego. Praktyczna obsługa kompilatora pod linuxem i MS Windows (np. g++, clang lub MS Visual Studio 2015).

## 5 EFEKTY KSZTAŁCENIA

- EK1** Wiedza: Potrafi przygotować projekt rozwiązujący dane zagadnienie w sposób obiektowy, wraz z wizualizacją na diagramach klas (UML), oraz zna wzorce projektowe, które mogą się przydać w rozwiązaniu. Projekt według opisu koduje (w języku C++).
- EK2** Wiedza: Zna bibliotekę STL, mechanikę iteratorów oraz wykorzystuje algorytmy z biblioteki standardowej. Posiada wiedzę o tym jak działają obiekty funkcyjne oraz wyrażenia lambda.
- EK3** Umiejętności: Potrafi napisać kod obsługujący sytuacje wyjątkowe. Poszukuje i naprawia błędy w kodzie za pomocą asercji, a także narzędzi diagnostycznych (debuger, profiler). Pisze i wykonuje testy (np. wg. CPPUNIT i wzorca kompozyt).
- EK4** Umiejętności: Wykonuje analizę kodu i przepisuje wg. reguł refaktoryzacji. Posiada umiejętność zapisania kodu działającego wielowątkowo.

## 6 TREŚCI PROGRAMOWE

### WYKŁAD

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Repetitorium z języka C++ z naciskiem na efektywne wykorzystanie standardu (C++14) i biblioteki standardowej oraz technik programowania obiektowego. Zasady wizualizacji zależności obiektowych i działania programu (diagramy klas i diagramy użycia UML).	3
W2	Przegląd wzorców projektowych i praktyczne przykłady ich zastosowań wraz z implementacją oraz wizualizacją w UML.	2
W3	Iteratory oraz kontenery standardowej biblioteki STL. Cechy kontenerów i wykorzystanie ich w algorytmach. Przegląd i zastosowanie algorytmów.	3
W4	Obiekty funkcyjne, ich współpraca z algorytmami. Wyrażenia lambda.	2
W5	Programowanie szablonami (szablony funkcji i szablony klas). Klasy cech, funkcje wytycznych, klasy wytycznych. Metaprogramowanie.	2
W6	Operacje atomowe. Programowanie wielowątkowe (wątki, operacje asynchroniczne itd).	3
	RAZEM	15

### PROJEKT

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P1	Projektowanie z wizualizacją w diagramach UML wybranego przypadku obiektowego, korzystającego z relacji dziedziczenia (również wielokrotnego) i relacji zawierania. Programowanie i testowanie kodu. Uzupełnienie wiadomości o nowych możliwościach języka C++ w standardzie C++14.	5
P2	Implementacja wybranych wzorców projektowych (wzorec obserwator, dekorator, fabryka abstrakcyjna, most, singleton).	5
P3	Ćwiczenia z wykorzystaniem algorytmów, kontenerów standardowych i iteratorów biblioteki standardowej. Wzorec iteratora.	4



## PROJEKT

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P4	Alokowanie pamięci. Inteligentne wskaźniki (unique_ptr, shared_ptr, weak_ptr). Zapobieganie wyciekom pamięci. Ćwiczenia z diagnozowania błędów w kodzie za pomocą asercji oraz narzędzi typu debugger oraz profiler.	4
P5	Technika pisania obiektów funkcyjnych oraz akomodowania istniejących kodów (metod) za pomocą więzadeł. Wyrażenia lambda jako przykład funktorów pisanych w miejscu wykonania.	4
P6	Programowanie szablonami. Klasy cech i wytycznych, funkcje typów. Metaprogramowanie i szablony wyrażeń. Szablony o zmiennej liczbie parametrów (variadic templates) i korzyści z nich wypływające.	4
P7	Model pamięci i operacje atomowe i programowanie współbieżne. Zarządzanie wątkami, problemy współdzielenia danych przez wątki. Synchronizacja operacji współbieżnych. Stosowanie blokad.	4
	RAZEM	30

## 7 METODY DYDAKTYCZNE

M1 Wykłady

M2 Ćwiczenia projektowe

M3 Praca w grupach

M4 Prezentacje multimedialne

M5 Studium przypadku

M6 Projekty

M7 Konsultacje

## 8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
<b>Godziny kontaktowe z nauczycielem akademickim, w tym:</b>	
Godziny wynikające z planu studiów	45
Konsultacje przedmiotowe	5
Egzaminy i zaliczenia w sesji	5
<b>Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:</b>	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	35
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	35
<b>SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA</b>	<b>125</b>
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	5

## 9 SPOSOBY OCENY

Ewaluacja zdobytych umiejętności w oparciu o wiedzę (sprawdzaną na egzaminie) oraz praktyczne umiejętności (sprawdzane poprzez kolokwia oraz projekty do domu).

**OCENA FORMUJĄCA****F1** Kolokwium**F2** Projekt indywidualny**F3** Referat**KRYTERIA OCENY**

EFEKT KSZTAŁCENIA 1		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 1
NA OCENĘ 3	Student potrafi, z drobnymi mankamentami, naszkicować rozwiązanie problemu na diagramach UML oraz zaprogramować je, choć wykazuje jeszcze brak wprawy i niekompletną wiedzę na temat np. dziedziczenia wielobazowego i wirtualnego dziedziczenia.	projekt	Projekt do wykonania indywidualnie (poza ćwiczeniami) i omówienie go z prowadzącym. Prezentacja wybranych projektów w formie mini-referatów.
NA OCENĘ 4	Student potrafi narysować diagramy klas UML, przedstawiając na nich swoje obiektowe rozwiązanie. Implementuje to rozwiązanie, ale z drobnymi problemami jeśli chodzi o umiejętność posługiwania się technikami obiektowymi w języku C++.		
NA OCENĘ 5	Student zna zasady zapisu projektu na diagramach UML (diagramy klas i diagramy użycia). Potrafi zaprojektować obiektowe rozwiązanie, a następnie je przedyskutować i zaimplementować w języku C++. Zna bardzo dobrze techniczne aspekty tworzenia klas i ich składowych, oraz dziedziczenia (w tym dziedziczenia wielobazowego).		
EFEKT KSZTAŁCENIA 2		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 2
NA OCENĘ 3	Student zna i potrafi napisać przykładowe proste implementacje obiektu funkcyjnego i wyrażenia lambda. Zna podstawowe kontenery, choć nie wszystkie ich szczegóły. Zasadniczo radzi sobie z wykorzystaniem iteratorów.	projekt	Kolokwium praktyczne podczas zajęć, polegające na indywidualnym rozwiązaniu krótkich zadań programistycznych.
NA OCENĘ 4	Student potrafi napisać obiekt funkcyjny lub zaadaptować istniejące metody, tak żeby użyć w ramach algorytmów. Korzysta z wyrażenia lambda. Zna i używa kontenerów standardowych wraz z iteratorami, ale nie posiada na ich temat kompletnej wiedzy.		
NA OCENĘ 5	Student wykazuje wiedzę na temat pisania obiektów funkcyjnych i wyrażenia lambda oraz zastosowania ich w algorytmach biblioteki standardowej. Swobodnie orientuje się i wykorzystuje kontenery standardowe, wie jak używać iteratorów (strumieniowych, oraz tych wykorzystywanych w kontenerach).		



EFEKT KSZTAŁCENIA 3		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 3
NA OCENĘ 3	Student szuka błędów za pomocą asercji oraz prostych działań z programem typu debugger. Potrafi omówić zasady testowania kodu, choć implementuje je w sposób nieoptymalny.	projekt	Krótkie kolokwium sprawdzające umiejętności naprawiania wadliwie napisanego kodu.
NA OCENĘ 4	Student potrafi szukać i usuwać błędy w kodzie, używając do tego celu debugera. Zna zasady pisania testów i potrafi je wdrożyć.		
NA OCENĘ 5	Student swobodnie posługuje się debuggerem a także wie na czym polega optymalizowanie i profilowanie kodu. Potrafi napisać zestaw testów (w oparciu o CPPUNIT, asercje).		
EFEKT KSZTAŁCENIA 4		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 4
NA OCENĘ 3	Student wie jak napisać program działający na wątkach. Zna reguły unikania konfliktów współdzielenia zasobów i stosowania odpowiednich blokad, ale nie posługuje się swobodnie bardziej zaawansowanymi technikami.	projekt	Projekt do pracy indywidualnej, ilustrujący zagadnienie programowania współbieżnego.
NA OCENĘ 4	Implementuje operacje atomowe i na wątkach, z pewnymi mankamentami potrafi zarządzać współdzielonymi zasobami.		
NA OCENĘ 5	Student potrafi przeanalizować kod pod kątem użyteczności implementacji go w postaci wielowątkowej. Potrafi napisać asynchroniczną obsługę wyjątków, wie na czym polega unikanie konfliktu współdzielenia zasobów, stawiania blokad itd.		

#### OCENA DO INDEKSU (OCENA PODSUMOWUJĄCA)

Ocena końcowa jest średnią arytmetyczną (czyli z wagą 0.5) oceny uzyskanej na końcowym egzaminie pisemnym oraz z zaliczenia zajęć projektowych.

#### WARUNKI ZALICZENIA PRZEDMIOTU

- a Pozytywne zaliczenie wszystkich etapów zajęć projektowych (czyli kolokwiów oraz prac domowych). Pozytywne zaliczenie egzaminu pisemnego.

## 10 MACIERZ REALIZACJI PRZEDMIOTU



EFEKTY KSZTAŁCENIA DLA PRZEDMIOTU	ODNIESIENIE DO EFEKTÓW KIERUNKOWYCH	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	METODY DYDAKTYCZNE
EK1	INF_UO02, INF_UO04, INF_UB08, INF_W07	Cel1	W1, W2, P1, P2	M1, M2, M3, M4, M6, M7
EK2	INF_UO02, INF_W07	Cel2, Cel3	W3, W4, P3, P4	M1, M2, M3, M5, M7
EK3	INF_UB02, INF_W07, INF_UB07	Cel4	W5, P5, P6	M1, M2, M4, M7
EK4	INF_W15, INF_W12, INF_UB08, INF_W07	Cel5	W6, P7	M1, M2, M4, M7

## 11 WYKAZ LITERATURY

### LITERATURA PODSTAWOWA:

- [1] Bjarne Stroustrup — *Programowanie. Teoria i praktyka z wykorzystaniem C++*. Wydanie II, Polska, 2013, Helion
- [2] Nicholas A. Solter, Scott J. Kepler — *C++ Zaawansowane programowanie*, Polska, 2005, Helion
- [3] Siddhartha Rao — *C++. Dla każdego*. Wydanie VII, Polska, 2013, Helion
- [4] Stephen Prata — *Język C++. Szkoła programowania*. Wydanie VI, Polska, 2013, Helion
- [5] Anthony Williams — *Język C++ i przetwarzanie współbieżne w akcji*, Polska, 2013, Helion
- [6] Nicolai M. Josuttis — *C++. Biblioteka standardowa. Podręcznik programisty*. Wydanie II, Polska, 2014, Helion
- [7] David Vandervoorde, Nicolai M. Josuttis — *C++ szablony. Vademecum profesjonalisty*, Polska, 2003, Helion
- [8] Scott Meyers — *Overview of The New C++ (C++11/14)*, USA, 2013, Artima
- [9] Scott Meyers — *Skuteczny nowoczesny C++. 42 sposoby lepszego posługiwania się językami C++11 i C++14*, Warszawa, 2015, Helion

### LITERATURA UZUPEŁNIAJĄCA:

- [1] Jerzy Grębosz — *Symfonia C++ Standard*, Polska, 2010, Editions 2000
- [2] Bruce Eckel — *Thinking in C++. Edycja polska*, Polska, 2002, Helion
- [3] D. Ryan Stephens — *C++ Receptury*, Polska, 2006, Helion
- [4] Harvey M. Deitel, Paul J. Deitel — *C++ How to Program, 10th Edition*, USA, 2014, Deitel
- [5] Bjarne Stroustrup — *Język C++. Kompendium wiedzy*, Warszawa, 2014, Helion



## 12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

### OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr Witold Przygoda (kontakt: witold.przygoda@gmail.com)

### OSOBY PROWADZĄCE PRZEDMIOT

dr Witold Przygoda (kontakt: witold.przygoda@gmail.com)

## 13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)	(odpowiedzialny za przedmiot)	(kierownik zakładu)	(dyrektor instytutu)
---------------------	-------------------------------	---------------------	----------------------

PWSZ w Nowym Sączu

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....