

KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Zaawansowane techniki programowania aplikacji sieciowych w języku C++ D1_12
Nazwa przedmiotu (j. ang.):	Advanced Paradigms of Network Application Development in C++ Language.
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	Informatyka praktyczna
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne / studia niestacjonarne
Obszar kształcenia:	nauki techniczne
Dziedzina:	nauki techniczne
Dyscyplina nauki:	Informatyka
Koordynator przedmiotu:	dr inż. Piotr Wais

2. Ogólna charakterystyka przedmiotu

Przynależność do modułu:	specjalizacyjnego do wyboru
Status przedmiotu:	obowiązkowy
Język wykładowy:	polski
Rok studiów, semestr:	II, 4
Forma i wymiar zajęć według planu studiów:	stacjonarne - wykład 15 h, ćw. laboratoryjne 30 h niestacjonarne - wykład 15 h, ćw. laboratoryjne 15 h
Interesariusze i instytucje partnerskie (nieobowiązkowe)	
Wymagania wstępne / Przedmioty wprowadzające:	Programowanie w języku C++ / Programowanie II

3. Bilans punktów ECTS

Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt =25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami):	6 (A + B)		
		stacjonarne	Niestacjonarne
A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontaktowych, w czasie rzeczywistym, w tym testy, egzaminy etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiągniętych na tych zajęciach	obecność na wykładach	15	15
	obecność na ćwiczeniach laboratoryjnych	30	15
	egzamin	2	2
	udział w konsultacjach	5	5
	w sumie:	52	37
	ECTS	2	1,5
B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS (np. praca w bibliotece, w sieci, na platformie e-learningowej, w laboratorium, praca nad projektem końcowym, przygotowanie ogólne; suma poszczególnych godzin powinna zgadzać się z liczbą ogólną)	przygotowanie do ćwiczeń laboratoryjnych	20	20
	wykonanie projektu	20	20
	praca w sieci	15	20
	przygotowanie do egzaminu	15	15
	przygotowanie do konsultacji	10	10
	uzupełnienie/studiowanie notatek	5	8
	studiowanie zalecanej literatury	15	20
	w sumie:	100	113
ECTS	4	4,5	
C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS (ta liczba nie musi być powiązana z liczbą godzin kontaktowych, niektóre zajęcia praktyczne/laboratoryjne mogą odbywać się bez udziału nauczyciela):	udział w ćwiczeniach laboratoryjnych	30	15
	wykonanie projektu	20	20
	praca praktyczna samodzielna	12	27
	w sumie:	62	62
	ECTS	2,5	2,5

4. Opis przedmiotu

<p>Cel przedmiotu: Osiągnięcie podstawowej wiedzy z zakresu programowania aplikacji sieciowych w środowisku MS Visual Studio C++.</p>
<p>Metody dydaktyczne: wykład informacyjny, pokaz, ćwiczenia laboratoryjne, metoda projektów</p> <p><i>np. podające (wykład), problemowe (konwersatorium, seminarium), aktywizujące (symulacja, metoda przypadków itp.), eksponujące (pokaz, film), praktyczne (ćwiczenia, metoda projektów itp) – pełniejszy wykaz poniżej (prosimy wybrać najstosowniejsze - jedną lub więcej, można dodać własne metody)</i></p>
<p>Treści kształcenia (w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach):</p> <p>Wykłady:</p> <ol style="list-style-type: none"> Visual Studio C++ - Platforma .NET – przegląd komponentów biblioteki Windows Forms, projektowanie interfejsu aplikacji, menu główne, menu kontekstowe, metody zdarzeniowe, okna dialogowe i pliki tekstowe, edycja i korzystanie ze schowka, drukowanie, ekran powitalny, przygotowanie ikony w obszarze powiadamiania, odtwarzanie pliku dźwiękowego, wczytywanie obrazu, ustawienia aplikacji. Programowanie sieciowe – sieci komputerowe, protokoły TCP i UDP, protokół IP i adresy MAC, programowanie klient- serwer i peer-to-peer. Aplikacje TCP i UDP - Połączenie TCP – klient, serwer, odczytanie adresu IP przyłączonego hosta.

Połączenie UDP – klient, serwer. Asynchroniczne połączenie TCP. Prosty skaner otwartych portów hosta zdalnego. Skaner otwartych portów lokalnego hosta. Sprawdzenie adresu IP naszego komputera. Komplet informacji na temat połączeń sieciowych. Ping. Ping - przeciwdziałanie zablokowaniu interfejsu. NetDetect - sprawdzanie dostępnych komputerów w sieci. Traceroute - śledzenie drogi pakietu ICMP.

4. Remoting - Serwer i klient TCP.

5. ASP.NET – ping, wysyłanie wiadomości e-mail, pobieranie plików na serwer

6. Web Services – usługa sieciowa

7. WCF - Podstawy działania. WCF = A + B + C. Definiowanie kontraktu. Udostępnianie usługi. Tworzenie klienta.

Ćwiczenia laboratoryjne:

1. Projektowanie aplikacji w środowisku Visual Studio C++ - Platforma .NET – komponenty biblioteki Windows Forms

2. Projektowanie aplikacji w środowisku Visual Studio C++:

- Aplikacje TCP i UDP,
- Remoting - Serwer i klient TCP.
- ASP.NET – ping, wysyłanie wiadomości e-mail, pobieranie plików na serwer
- Web Services – usługa sieciowa
- WCF - Definiowanie kontraktu. Udostępnianie usługi. Tworzenie klienta.

5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia (w sumie wymienić ok. od 3 do 9 efektów - podać numery efektów z listy dla danego kierunku/specjalności – opublikowane na stronie uczelni; podać TYLKO te efekty (tam gdzie to możliwe i stosowne w trzech kategoriach, np. kompetencje społeczne mogą nie być realizowane w tym przedmiocie), na których osiągnięcie kładzie się nacisk w ramach przedmiotu, wybrane efekty kierunkowe powinny być bardziej szczegółowo sformułowane niż te dla całej specjalności, tak aby były weryfikowalne – dlatego mają osobne symbole jako efekty przedmiotu)

Efekt przedmiotu (kod przedmiotu + kod efektu kształcenia)	Student, który zaliczył przedmiot (spełnił minimum wymagań)	Efekt kierunkowy
D1_12_W01 D1_12_W02 D1_12_W03	Wiedza: <ol style="list-style-type: none"> 1. Student zna narzędzia i mechanizmy potrzebne do zbudowania aplikacji Windows Forms w środowisku MS Visual Studio C++. 2. Student potrafi programować aplikacje w środowisku MS Visual Studio C++ wykorzystując technologie sieciowe. 3. Student wie jak programować dostosowując swój projekt do ciągle zmieniających się trendów i możliwości. 	K_W08 K_W16 K_W07
D1_12_U01 D1_12_U02 D1_12_U03	Umiejętności: <ol style="list-style-type: none"> 1. Student potrafi poszerzać i aktualizować swoją wiedzę niezbędną do zbudowania aplikacji w środowisku MS Visual Studio C++ wykorzystując komponenty biblioteki Windows Forms zgodnie z obowiązującymi standardami i rozwiązaniami. 2. Student potrafi na podstawie algorytmu (specyfikacji) stworzyć prostą aplikację sieciową w środowisku MS Visual Studio C++ wykorzystując technologie sieciowe. 3. Student potrafi zarządzać danymi z poziomu aplikacji. 	K_U03 K_U04 K_U10 K_U17 K_U18 K_U11

D1_12_K01	Kompetencje społeczne 1. Student rozumie potrzebę poznawania nowych narzędzi programistycznych wykorzystywanych w środowisku MS Visual Studio C++.	K_K01
D1_12_K02		2. Student potrafi pracując w zespole zaprojektować i zaimplementować aplikację sieciową w środowisku MS Visual Studio C++.

Sposoby weryfikacji efektów kształcenia:

(np. dyskusja, gra dydaktyczna, zadanie e-learningowe, ćwiczenie laboratoryjne, projekt indywidualny/grupowy, zajęcia terenowe, referat studenta, praca pisemna, kolokwium, test zaliczeniowy, egzamin, opinia eksperta zewnętrznego, etc. Dodać do każdego wybranego sposobu symbol zakładanego efektu, jeśli jest ich więcej)

Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca	Ocena końcowa
1	D1_12_W01 D1_12_W02 D1_12_W03 D1_12_U01 D1_12_U02 D1_12_U03	Egzamin	ocena z egzaminu – sprawdzian wiedzy i umiejętności	Ocena końcowa z egzaminu
2	D1_12_U01 D1_12_U02 D1_12_U03 D1_12_K01 D1_12_K02	ćwiczenia laboratoryjne, projekt indywidualny/grupowy	ocena z projektu końcowego, ocena zaangażowania na zajęciach	Ocena końcowa z laboratorium - średnia z ocen formujących sprawdzających nabyte umiejętności i kompetencje

Kryteria oceny (oceny 3,0 powinny być równoważne z efektami kształcenia, choć mogą być bardziej szczegółowo opisane):

w zakresie wiedzy		Efekt kształcenia
Na ocenę 3,0	Student uzyskał min. 50% wymaganej wiedzy w zakresie obowiązującego materiału. Student: - zna podstawowe narzędzia potrzebne do zbudowania aplikacji Windows Forms – MS Visual Studio C++, oraz mechanizmy, takie jak: kreator projektu, - zna podstawowe pojęcia w zakresie programowania aplikacji sieciowych w środowisku MS Visual Studio C++ wykorzystując technologie sieciowe. - wie jak aktualizować środowisko programistyczne potrzebne do programowania aplikacji w obecnie obowiązującej wersji.	D1_12_W01 D1_12_W02 D1_12_W03
Na ocenę 5,0	Student zdobył powyżej 95% wymaganej wiedzy w zakresie obowiązującego materiału. Student: - zna dodatkowe mechanizmy, takie jak: debugger, - zna podstawowe i zaawansowane pojęcia w zakresie programowania aplikacji sieciowych w środowisku MS Visual Studio C++ wykorzystując technologie sieciowe	D1_12_W01 D1_12_W02 D1_12_W03

	- zna sposoby wykorzystania dodatkowych funkcji środowiska programistycznego.	
w zakresie umiejętności		
Na ocenę 3,0	<p>Student uzyskał min. 50% wymaganych umiejętności w zakresie obowiązującego materiału. Student potrafi:</p> <ul style="list-style-type: none"> - stworzyć nowy projekt zgodnie z obowiązującymi standardami i rozwiązaniami. - na podstawie algorytmu (specyfikacji) stworzyć prostą aplikację w środowisku MS Visual Studio C++ wykorzystując komponenty biblioteki Windows Forms. - stworzyć prostą aplikację sieciową w środowisku MS Visual Studio C++ wykorzystując technologie sieciowe. 	<p>D1_12_U01</p> <p>D1_12_U02</p> <p>D1_12_U03</p>
Na ocenę 5,0	<p>Student uzyskał powyżej 95% umiejętności w zakresie obowiązującego materiału. Student umie:</p> <ul style="list-style-type: none"> - stworzyć nowy projekt zgodnie z obowiązującymi standardami i rozwiązaniami oraz dołączyć dodatkowe biblioteki do projektu - na podstawie algorytmu (specyfikacji) stworzyć zaawansowaną aplikację w środowisku MS Visual Studio C++ wykorzystując komponenty biblioteki Windows Forms. - stworzyć zaawansowaną aplikację sieciową w środowisku MS Visual Studio C++ wykorzystując technologie sieciowe. 	<p>D1_12_U01</p> <p>D1_12_U02</p> <p>D1_12_U03</p>
w zakresie kompetencji społecznych		
Na ocenę 3,0	Student osiągnął wymagane kompetencje społeczne na poziomie min. 50%.	<p>D1_12_K01</p> <p>D1_12_K02</p>
Na ocenę 5,0	Student osiągnął wymagane kompetencje społeczne na poziomie wyższym niż 90%.	<p>D1_12_K01</p> <p>D1_12_K02</p>
<p>Kryteria oceny końcowej (zaleca się podział procentowy poszczególnych kryteriów składających się na ocenę końcową, który może współgrać z powyższymi kryteriami: np. aktywność za zajęciach.. %, kolokwia ...%, samodzielne ćwiczenia ...%, laboratoria ... % ocena z projektu (szczególnie istotna)- ...%, zajęcia terenowe...%, zaliczenie, egzamin pisemny... %, opinia eksperta zewnętrznego ...% itp.)</p> <p>ocena z egzaminu: 100%</p> <p>ocena z laboratorium: ocena z projektu: 80 % samodzielne wykonanie ćwiczeń laboratoryjnych: 15% aktywność na zajęciach: 5%</p>		
Zalecana literatura		
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. Matulewski J., Visual Studio 2013: podręcznik programowania w C# z zadaniami 3. Maciej Grabek, WCF od podstaw. Komunikacja sieciowa nowej generacji, Helion 2012 4. msdn.microsoft.com 5. cnap.pwsz.krosno.pl <p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. Sharp, John, Microsoft Visual C# 2013: Krok po kroku, APN Promise, 2014 		

Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:
Przygotowanie do wykładów i ćwiczeń laboratoryjnych – 30 godzin
Konsultacje – 10 godzin
Poprawa prac projektowych – 15 godzin
Przygotowanie i poprawa egzaminu – 10 godzin
W sumie: 65 godzin