

# KARTA PRZEDMIOTU

## 1. Informacje ogólne

<b>Nazwa przedmiotu i kod (wg planu studiów):</b>	Programowanie II C16
<b>Nazwa przedmiotu (j. ang.):</b>	Programming II
<b>Kierunek studiów:</b>	Informatyka
<b>Specjalność/specjalizacja:</b>	Sieciowe Systemy Informatyczne / Technologie internetowe i bazy danych / Informatyka praktyczna
<b>Poziom kształcenia:</b>	studia I stopnia
<b>Profil kształcenia:</b>	praktyczny (P)
<b>Forma studiów:</b>	studia stacjonarne / studia niestacjonarne
<b>Obszar kształcenia:</b>	nauki techniczne
<b>Dziedzina:</b>	nauki techniczne
<b>Dyscyplina nauki:</b>	Informatyka
<b>Koordinator przedmiotu:</b>	dr inż. Piotr Wais

## 2. Ogólna charakterystyka przedmiotu

<b>Przynależność do modułu:</b>	kształcenia kierunkowego
<b>Status przedmiotu:</b>	obowiązkowy
<b>Język wykładowy:</b>	polski
<b>Rok studiów, semestr:</b>	II, 3
<b>Forma i wymiar zajęć według planu studiów:</b>	stacjonarne - wykład 30 h, ćw. laboratoryjne 30 h niestacjonarne - wykład 15 h, ćw. laboratoryjne 15 h
<b>Interesariusze i instytucje partnerskie (nieobowiązkowe)</b>	
<b>Wymagania wstępne / Przedmioty wprowadzające:</b>	Język C/Programowanie I

### 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): (A + B)	5		
		stacjonarne	Niestacjonarne
<b>A. Liczba godzin wymagających bezpośredniego udziału nauczyciela</b> (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	30	15
	obecność na ćwiczeniach laboratoryjnych	30	15
	egzamin	2	2
	udział w konsultacjach	5	5
	<b>w sumie:</b> ECTS	67 2,6	37 1,4
<b>B. Poszczególne typy zadań do samokształcenia studenta</b> (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	10	10
	wykonanie sprawozdań	10	15
	przygotowanie do kolokwium	10	15
	praca w sieci	5	10
	przygotowanie do egzaminu	10	10
	przygotowanie do konsultacji	5	10
	uzupełnienie/studiowanie notatek	5	10
	studiowanie zalecanej literatury	5	10
	<b>w sumie:</b> ECTS	60 2,4	90 3,6
	<b>C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS</b> (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
praca praktyczna samodzielna		30	45
<b>w sumie:</b> ECTS		60 2	60 2

### 4. Opis przedmiotu

<p><b>Cel przedmiotu:</b> Osiągnięcie podstawowej wiedzy z zakresu programowania obiektowego w języku C++ i języku C#</p>
<p><b>Metody dydaktyczne:</b> wykład informacyjny, pokaz, ćwiczenia laboratoryjne</p>
<p><b>Treści kształcenia</b> (w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach):</p> <p><b>Wykłady:</b></p> <ol style="list-style-type: none"> <li>1. Wprowadzenie do programowania obiektowego w języku C++. Deklaracja i czas trwania obiektów. Klasy: deklaracja i definicja klasy, składniki klasy, enkapsulacja, hermetyzacja,</li> <li>2. Klasy: funkcje składowe, wskaźnik this, przesłanianie nazw zmiennych i funkcji, przeładowanie nazw funkcji, argumenty domyślne. Przekazywanie obiektów do funkcji i zwracanie obiektu przez funkcje. Składnik statyczny klasy. Statyczna funkcja składowa klasy. Funkcje zaprzyjaźnione. Zaprzyjaźnienie klas.</li> <li>3. Konstruktory i destruktory. Konstruktor domniemany. Lista inicjalizacyjna konstruktora. Konstruktor kopiujący.</li> <li>4. Przeładowanie operatorów. Funkcja operatorowa składowa klasy. Operatory predefiniowane. Argumentowość operatorów. Przeładowanie operatora przypisania =. Przeładowanie operatorów &lt;&lt; i &gt;&gt; dla strumieni wejścia/wyjścia.</li> <li>5. Dynamiczne alokowanie pamięci. Operatory new, delete.</li> </ol>

6. Dziedziczenie: istota dziedziczenia, dostęp do składników. Dziedziczenie kilkupokoleniowe. Przypisanie i inicjalizacja obiektów w warunkach dziedziczenia. Dziedziczenie wielokrotne. Klasy wirtualne.
7. Funkcje wirtualne. Polimorfizm. Klasy abstrakcyjne. Szablony funkcji i klas.
8. Operacje wejścia/wyjścia: strumień, operacje we/wy na plikach.
9. Programowanie obiektowe w języku C# - Platforma .NET - środowisko uruchomieniowe, podstawowe typy danych, metody, typy wartościowe i referencyjne, typy dynamiczne, sterowanie przepływem, wyjątki, dyrektywy preprocesora.
10. Programowanie obiektowe w języku C# - klasy, pola, metody, hermetyzacja, modyfikatory dostępu.
11. Programowanie obiektowe w języku C# - dziedziczenie, hierarchia klas.
12. Programowanie obiektowe w języku C# - konstruktor, destruktor, właściwości, elementy statyczne, polimorfizm, abstrakcja.
13. Programowanie obiektowe w języku C# - interfejsy, delegaty i zdarzenia.

### **Ćwiczenia laboratoryjne:**

1. Wprowadzenie do programowania w języku C++. Pierwsze programy. Środowisko uruchomieniowe. Deklaracja i definicja klasy. Dane klasy. Enkapsulacja.
2. Funkcje składowe klasy. Przesłanianie nazw zmiennych i funkcji. Przeładowanie nazw funkcji, argumenty domyślne. Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez referencję). Zwrocenie obiektu przez funkcję. Składnik statyczny klasy. Statyczna funkcja składowa klasy. Funkcje zaprzyjaźnione do klasy.
3. Konstruktory i destruktory. Inicjowanie i niszczenie obiektu. Konstruktor kopiujący.
4. Przeładowanie operatora przypisania =. Przeładowanie operatorów << i >> dla strumieni wejścia/wyjścia.
5. Operatory new, delete. Dynamiczne alokowanie pamięci.
6. Wprowadzenie w dziedziczenie, hierarchia klas.
7. Funkcje wirtualne. Polimorfizm. Klasy abstrakcyjne.
8. Programowanie obiektowe w języku C# - Platforma .NET - środowisko uruchomieniowe, aplikacja konsolowa, podstawowe typy danych, metody, typy wartościowe i referencyjne, typy dynamiczne, sterowanie przepływem, wyjątki, dyrektywy preprocesora.
9. Programowanie obiektowe w języku C# - klasy, pola, metody, hermetyzacja, modyfikatory dostępu.
10. Programowanie obiektowe w języku C# - dziedziczenie, hierarchia klas.
11. Programowanie obiektowe w języku C# - konstruktor, destruktor, właściwości, elementy statyczne, polimorfizm, abstrakcja.
12. Programowanie obiektowe w języku C# - interfejsy, delegaty i zdarzenia.

## 5. Efekty kształcenia i sposoby weryfikacji

<b>Efekty kształcenia</b> (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)				
<b>Efekt przedmiotu</b> (kod przedmiotu + kod efektu kształcenia)	<b>Student, który zaliczył przedmiot (spełnił minimum wymagań)</b>			<b>Efekt kierunkowy</b>
C16_W01 C16_W02 C16_W03	<b>Wiedza:</b> 1. Student zna narzędzia i mechanizmy potrzebne do zbudowania aplikacji w języku programowania C++ i C#. 2. Student wie jak programować aplikacje wykorzystując techniki programowania obiektowego w językach C++ i C#. 3. Student wie jak programować dostosowując swój projekt do ciągle zmieniających się trendów i możliwości.			K_W08  K_W16  K_W07
C16_U01  C16_U02  C16_U03	<b>Umiejętności:</b> 1. Student potrafi poszerzać i aktualizować swoją wiedzę niezbędną do zbudowania aplikacji w języku programowania C++ i C# zgodnie z obowiązującymi standardami i rozwiązaniami.  2. Student potrafi na podstawie algorytmu (specyfikacji) stworzyć prostą aplikację wykorzystując techniki programowania obiektowego w językach C++ i C#.  3. Student potrafi zarządzać danymi z poziomu aplikacji			K_U03  K_U10  K_U11
C16_K01  C16_K02	<b>Kompetencje społeczne</b> 1. Student rozumie potrzebę poznawania nowych narzędzi programistycznych wykorzystywanych w języku programowania C++ i C#.  2. Student rozumie potrzebę wykorzystania nabytej wiedzy na niezwykle szybko rozwijającym się rynku aplikacji.			K_K01  K_K08
<b>Sposoby weryfikacji efektów kształcenia:</b> <i>(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)</i>				
Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca	Ocena końcowa
1	C16_W01 C16_W02 C16_W03 C16_U01 C16_U02 C16_U03	Egzamin	ocena z egzaminu – sprawdzian wiedzy i umiejętności	Ocena końcowa z egzaminu

2	C16_W01 C16_W02 C16_W03 C16_U01 C16_U02 C16_U03	kolokwium zaliczeniowe	ocena z kolokwium - sprawdzian wiedzy i umiejętności	Ocena końcowa z laboratorium - średnia z ocen formujących
3	C16_U01 C16_U02 C16_U03 C16_K01 C16_K02	ćwiczenia laboratoryjne	ocena sprawozdania z prac laboratoryjnych, ocena zaangażowania na zajęciach	

**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 w języku programowania C++ i C# dla SO Windows – Visual Studio, oraz mechanizmy, takie jak: kreator projektu, - wie jak programować aplikacje wykorzystując podstawowe techniki programowania obiektowego w językach C++ i C#, - wie jak aktualizować środowisko programistyczne potrzebne do programowania aplikacji w obecnie obowiązującej wersji.	C16_W01  C16_W02  C16_W03	
Na ocenę 5,0	Student zdobył powyżej 95% wymaganej wiedzy w zakresie obowiązującego materiału. Student: - zna dodatkowe mechanizmy, takie jak: debugger, - wie jak programować aplikacje wykorzystując zaawansowane techniki programowania obiektowego w językach C++ i C# - takie jak: dziedziczenie - zna sposoby wykorzystania dodatkowych funkcji środowiska programistycznego.	C16_W01 C16_W02  C16_W03	
w zakresie umiejętności			
Na ocenę 3,0	Student uzyskał min. 50% wymaganych umiejętności w zakresie obowiązującego materiału. Student potrafi: - stworzyć nowy projekt zgodnie z obowiązującymi standardami i rozwiązaniami. - wykorzystać podstawowe techniki programowania obiektowego w językach C++ i C# aby na podstawie algorytmu (specyfikacji) stworzyć prostą aplikację konsolową - przetwarzać dane w aplikacji konsolowej	C16_U01  C16_U02  C16_U03	
Na ocenę 5,0	Student uzyskał powyżej 95% umiejętności w zakresie obowiązującego materiału. Student umie: - stworzyć nowy projekt zgodnie z obowiązującymi standardami i rozwiązaniami oraz dołączyć dodatkowe biblioteki do projektu - wykorzystać zaawansowane techniki programowania obiektowego w językach C++ i C# (np. dziedziczenie) aby na podstawie algorytmu (specyfikacji) stworzyć prostą aplikację konsolową - przetwarzać dane w aplikacji konsolowej stosując dynamiczny przydział pamięci	C16_U01  C16_U02  C16_U03	

<b>w zakresie kompetencji społecznych</b>		
Na ocenę 3,0	Student osiągną wymagane kompetencje społeczne na poziomie min. 50%.	C16_K01 C16_K02
Na ocenę 5,0	Student osiągną wymagane kompetencje społeczne na poziomie wyższym niż 90%.	C16_K01 C16_K02
<p><b>Kryteria oceny końcowej</b> (zaleca się podział procentowy poszczególnych kryteriów składających się na ocenę końcową, który może współgrać z powyższymi kryteriami: np. aktywność za zajęciami.. %, 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 kolokwium: 70 % ocena ze sprawozdania: 10% samodzielne wykonanie ćwiczeń laboratoryjnych: 15% aktywność na zajęciach: 5%</p>		
<p><b>Zalecana literatura</b></p> <p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"> <li>Grębosz J, Symfonia C++ Standard : Programowanie w języku C++ orientowane obiektowo. Tom I, II, III, Kraków, 2006, Editions</li> <li>Matulewski J., Borycki D., Krause G., Grabek M., Pakulski M., Warczak M., Lewandowski J., Orłowski S., Visual Studio 2010 dla programistów C#, Helion</li> <li>cnap.pwsz.krosno.pl</li> </ol> <p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"> <li>Stroustrup B., Język C++, Warszawa, WNT 2000.</li> </ol>		

### Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:
Przygotowanie do wykładów i ćwiczeń laboratoryjnych – 30 godzin
Konsultacje – 10 godzin
Poprawa sprawozdań z ćwiczeń laboratoryjnych – 10 godzin
Przygotowanie i poprawa kolokwium zaliczeniowego – 5 godzin
Przygotowanie i poprawa egzaminu – 10 godzin
W sumie: 65 godzin