KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Programowanie I **C6**
Nazwa przedmiotu (j. ang.):	Programming I
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	Sieciowe Systemy Informatyczne / Technologie internetowe i bazy danych / Informatyka praktyczna/Bezpieczeństwo systemów informatycznych
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne
Obszar kształcenia:	nauki techniczne
Dziedzina:	nauki techniczne
Dyscyplina nauki:	Informatyka
Koordynator przedmiotu:	dr Jolanta Wojtowicz

2. Ogólna charakterystyka przedmiotu

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

3. Bilans punktów ECTS

		stacjonarne
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	
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 osiąganych na tych zajęciach**	obecność na wykładach obecność na ćwiczeniach laboratoryjnych udział w konsultacjach **w sumie:** ECTS	30 30 5 65 2,5
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 wykonanie sprawozdań przygotowanie do kolokwium praca w sieci przygotowanie do konsultacji uzupełnienie/studiowanie notatek studiowanie zalecanej literatury **w sumie:** ECTS	15 15 15 5 5 5 5 65 2,5
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 praca praktyczna samodzielna **w sumie:** ECTS	30 30 60 2

4. Opis przedmiotu

Cel przedmiotu:

Osiągnięcie podstawowej wiedzy z zakresu programowania obiektowego w języku C++ .

Metody dydaktyczne: wykład informacyjny, pokaz, ćwiczenia laboratoryjne.

Treści kształcenia *(w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach):*

Wykłady:

1. Wprowadzenie do programowania obiektowego w języku C++. Podstawowe cechy i zastosowania, kompilatory. Różnice między językiem C a C++. Obsługa wejścia/ wyjścia. Przykłady prostych programów. Składnia i elementy języka: typy danych, zmienne, instrukcje sterujące przepływem danych w programie.

2. Operatory. Referencje. Funkcje. Dostęp do biblioteki C.

3. Przekazywanie argumentów do funkcji. Argumenty domniemane. Funkcje inline. Obiekty lokalne i globalne, obiekty automatyczne.

4. Praca ze wskaźnikami. Operator rzutowania reinterpret_cast, a wskaźniki. Zastosowanie wskaźników w argumentach funkcji. Rezerwacja obszarów pamnięci operatory new i delete. Dynamiczna alokacja tablicy.

5. Przeładowanie nazw funkcji. Przeładowanie a zakres ważności deklaracji funkcji. Adres funkcji przeładowanej.

6. Klasy: deklaracja i definicja klasy, składniki klasy, enkapsulacja, hermetyzacja. 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.

7. Konstruktory i destruktory. Konstruktor domniemany. Lista inicjalizacyjna konstruktora. Konstruktor kopiujący.

8. Przeładowanie operatorów. Funkcja operatorowa składowa klasy. Operatory predefiniowane. Argumentowość operatorów. Przeładowanie operatora przypisania =. Przeładowanie operatorów << i >> dla strumieni wejścia/wyjścia.

9. Dziedziczenie: istota dziedziczenia, dostęp do składników. Dziedziczenie kilkupokoleniowe. Przypisanie i inicjalizacja obiektów w warunkach dziedziczenia. Dziedziczenie wielokrotne. Klasy wirtualne.

10. Funkcje wirtualne. Polimorfizm. Klasy abstrakcyjne. Szablony funkcji i klas.

11. Operacje wejścia/wyjścia: strumień, operacje we/wy na plikach.

Ćwiczenia laboratoryjne:

1. Wprowadzenie do programowania w języku C++. Pierwsze programy. Środowisko uruchomieniowe.
2. Operatory. Referencje. Funkcje. Dostęp do biblioteki C. Definiowane własnych typów danych w języku C++.
3. Przekazywanie argumentów do funkcji. Argumenty domniemane. Funkcje inline. Obiekty lokalne i globalne, obiekty automatyczne.
4. Praca ze wskaźnikami. Operator rzutowania reinterpret_cast, a wskaźniki. Zastosowanie wskaźników w argumentach funkcji. Rezerwacja obszarów pamnięci operatory new i delete. Dynamiczna alokacja tablicy.
5. Przeładowanie nazw funkcji. Przeładowanie a zakres ważności deklaracji funkcji. Adres funkcji przeładowanej.
6. Deklaracja i definicja klasy. Dane klasy. Enkapsulacja, hermetyzacja.
7. Funkcje składowe klasy. Przesłanianie nazw zmiennych i funkcji. Przeładowanie nazw funkcji, argumenty domyślne. Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez referencję). Zwracanie obiektu przez funkcje. Składnik statyczny klasy. Statyczna funkcja składowa klasy. Funkcje zaprzyjaźnione do klasy.
8. Konstruktory i destruktory. Inicjowanie i niszczenie obiektu. Konstruktor kopiujący.
9. Przeładowanie operatora przypisania =. Przeładowanie operatorów << i >> dla strumieni wejścia/wyjścia.
10. Operatory new, delete. Dynamiczne alokowanie pamięci.
11. Wprowadzenie w dziedziczenie, hierarchia klas.
12. Funkcje wirtualne. Polimorfizm. Klasy abstrakcyjne.

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**
	Wiedza:	
C6_W01	**1.** Student zna narzędzia i mechanizmy potrzebne do zbudowania aplikacji w języku programowania C++ .	K_W08
C6_W02	**2.** Student wie jak programować aplikacje wykorzystując techniki programowania obiektowego w językach C++.	K_W16
C6_W03	**3.** Student wie jak programować dostosowując swój projekt do ciągle zmieniających się trendów i możliwości.	K_W07
	Umiejętności:	
C6_U01	**1.** Student potrafi poszerzać i aktualizować swoją wiedzę niezbędną do zbudowania aplikacji w języku programowania C++ zgodnie z obowiązującymi standardami i rozwiązaniami.	K_U03
C6_U02	**2.** Student potrafi na podstawie algorytmu (specyfikacji) stworzyć prostą aplikację wykorzystując techniki programowania obiektowego w językach C++.	K_U10
C6_U03	**3.** Student potrafi zarządzać danymi z poziomu aplikacji	K_U11
	Kompetencje społeczne	
C6_K01	**1.** Student rozumie potrzebę poznawania nowych narzędzi programistycznych wykorzystywanych w języku programowania C++.	K_K01
C6_K02	**2.** Student rozumie potrzebę wykorzystania nabytej wiedzy na niezwykle szybko rozwijającym się rynku aplikacji.	K_K08

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.	C6_W01 C6_W02 C6_W03 C6_U01 C6_U02 C6_U03	kolokwium zaliczeniowe	ocena z kolokwium - sprawdzian wiedzy i umiejętności	Ocena końcowa z laboratorium - średnia z ocen formujących

2	C6_U01 C6_U02 C6_U03 C6_K01 C6_K02	ćwiczenia laboratoryjne	ocena sprawozdania z prac laboratoryjnych, ocena z aktywności na zajęciach	Ocena końcowa z laboratorium - średnia z ocen formujących

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++ dla SO Windows – Dev C++ Visual Studio, oraz mechanizmy, takie jak: kreator projektu, - wie jak programować aplikacje wykorzystując podstawowe techniki programowania obiektowego w językach C++, - wie jak aktualizować środowisko programistyczne potrzebne do programowania aplikacji w obecnie obowiązującej wersji.	C6_W01 C6_W02 C6_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 zawansowane techniki programowania obiektowego w językach C++ - takie jak: dziedziczenie, polimorfizm - zna sposoby wykorzystania dodatkowych funkcji środowiska programistycznego.	C6_W01 C6_W02 C6_W03
w zakresie umiejętności		
Na ocenę 3,0	Student uzyskał min. 50% wymaganych umiejętności w zakresie obowiązującego materiału. Student potrafi: - tworzyć nowy projekt zgodnie z obowiązującymi standardami i rozwiązaniami. - wykorzystać podstawowe techniki programowania obiektowego w językach C++ i C# aby na postawie algorytmu (specyfikacji) stworzyć prostą aplikację konsolową - przetwarzać dane w aplikacji konsolowej	C6_U01 C6_U02 C6_U03
Na ocenę 5,0	Student uzyskał powyżej 95% umiejętności w zakresie obowiązującego materiału. Student umie: - tworzyć nowy projekt zgodnie z obowiązującymi standardami i rozwiązaniami oraz dołączyć dodatkowe biblioteki do projektu - wykorzystać zawansowane techniki programowania obiektowego w językach C++ (np. dziedziczenie) aby na postawie algorytmu (specyfikacji) stworzyć prostą aplikację konsolową - przetwarzać dane w aplikacji konsolowej stosując dynamiczny przydział pamięci	C6_U01 C6_U02 C6_U03
w zakresie kompetencji społecznych		
Na ocenę 3,0	Student osiągną wymagane kompetencje społeczne na poziomie min. 50%.	C6_K01

		C6_K02
Na ocenę 5,0	Student osiągną wymagane kompetencje społeczne na poziomie wyższym niż 90%.	C6_K01 C6_K02

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.)*

ocena z laboratorium:
ocena z kolokwium: 70 %
ocena ze sprawozdania: 10%
samodzielne wykonanie ćwiczeń laboratoryjnych: 15%
aktywność na zajęciach: 5%

Zalecana literatura

Literatura podstawowa:

1. Gręboszj, Symfonia C++ Standard : Programowanie w języku C++ orientowane obiektowo. Tom I, II, III, Kraków, 2006, Editions
2. Stroustrup B., Jezyk C++, Warszawa, WNT 2000.
3. cnap.pwsz.krosno.pl

Literatura uzupełniająca:

1. Eckel B., "Thinking in C++", Helion 2002 / Prentice Hall 2000.
2. Josuttis N., "C++. Programowanie zorientowane obiektowo", Helion 2003 / Object– Oriented Programming in C++, Wiley 2002.

Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:
Przygotowanie do wykładów i ćwiczeń laboratoryjnych – 35 godzin
Konsultacje – 10 godzin
Poprawa sprawozdań z ćwiczeń laboratoryjnych – 10 godzin
Przygotowanie i poprawa kolokwium zaliczeniowego – 10 godzin
W sumie: 65 godzin