

KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	Z-ID-304b
Nazwa modułu	Programowanie w środowisku RAD C++
Nazwa modułu w języku angielskim	Programming in RAD C++ Environment
Obowiązuje od roku akademickiego	2015/2016

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Inżynieria danych
Poziom kształcenia	I stopień
Profil studiów	Praktyczny
Forma i tryb prowadzenia studiów	Studia stacjonarne
Specjalność	Wszystkie specjalności
Jednostka prowadząca moduł	Katedra Informatyki i Matematyki Stosowanej
Koordinator modułu	Mgr Grażyna Gębal Dr hab. Marzena Nowakowska
Zatwierdził	

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Kierunkowy
Status modułu	Wybieralny
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	Semestr III
Usytuowanie realizacji przedmiotu w roku akademickim	Semestr zimowy
Wymagania wstępne	Podstawy informatyki, Logika
Egzamin (TAK/NIE)	TAK
Liczba punktów ECTS	6

Forma prowadzenia zajęć	wykład w	ćwiczenia ć	laboratorium l	projekt p	inne i
Liczba godzin w semestrze	24		30		

C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	<p>Wykształcenie umiejętności budowania programu komputerowego w zakresie obejmującym:</p> <ul style="list-style-type: none"> - instrukcje języka C++ i podstawowe struktury danych, - programowanie w środowisku graficznym systemu RAD C++ z wykorzystaniem funkcji obsługi zdarzeń, - zasady posługiwania się komponentami oferowanymi w paletach standardowych RAD C++. <p>Treści merytoryczne przedmiotu będą realizowane w korelacji z przedmiotem <i>Algorytmy i struktury danych</i>.</p>
-------------------	--

Symbol efektu	Efekty kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów kierunkowych	Odniesienie do efektów obszarowych
W_01	Student zna i rozumie zasady funkcjonowania programów komputerowych i zarządzania nimi w środowisku Windows.	w, l	K_W11	T1P_W04 T1P_W06 X1P_W04
W_02	Student ma podstawową wiedzę na temat typów danych i doboru tych typów do rozwiązania określonego zadania programistycznego.	w, l	K_W09 K_W10	T1P_W04 T1P_W06 X1P_W04
W_03	Student ma wiedzę w zakresie syntaktyki i semantyki instrukcji sterujących języka C++.	w, l	K_W09 K_W10	T1P_W04 T1P_W06 X1P_W04
W_04	Student rozumie budowę programu komputerowego oraz rolę funkcji wbudowanych i własnych w tym programie.	w, l	K_W10 K_W11	T1P_W04 T1P_W06 X1P_W04
W_05	Student zna podstawowe komponenty środowiska RAD C++ do budowania graficznego interfejsu programu komputerowego.	w, l	K_W10 K_W11	T1P_W04 T1P_W06 X1P_W04
W_06	Student ma podstawową wiedzę z zakresu diagnozowania błędów w programie komputerowym.	w, l	K_W10 K_W11	T1P_W04 T1P_W06 X1P_W04
U_01	Student posiada umiejętności konstruowania algorytmów służących rozwiązaniu różnych zadań programistycznych.	l	K_U09	T1P_U14 T1P_U15 T1P_U16
U_02	Student potrafi wykorzystać środowisko RAD C++ do zaprojektowania i zbudowania programu komputerowego, kompilacji, weryfikacji błędów, konsolidacji, testowania programu.	w, l	K_U09	T1P_U14 T1P_U15 T1P_U16
U_03	Student potrafi zaprojektować i zbudować program komputerowy, korzystając z gotowych obiektów systemu programistycznego RAD C++ oraz własnych rozwiązań programistycznych.	l	K_U09	T1P_U14 T1P_U15 T1P_U16
U_04	Student potrafi dokonać analizy prostego kodu źródłowego, zdiagnozować błędy oraz wprowadzić modyfikacje w treści istniejącego programu.	w, l	K_U09	T1P_U14 T1P_U15 T1P_U16
U_05	Student posiada umiejętności definiowania własnych funkcji programisty oraz wykorzystania ich w tworzonym przez siebie programie komputerowym.	l	K_U09	T1P_U14 T1P_U15 T1P_U16
K_01	Student rozumie potrzebę stałego uzupełniania i rozwijania wiedzy z zakresu programowania.	w, l	K_K01	T1P_K01 X1P_K01 X1P_K05
K_02	Student potrafi pracować samodzielnie i w grupie (przyjmując w niej różne role).	l	K_K04	T1P_K03 T1P_K04 X1P_K02

Treści kształcenia

1. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1-2	Środowisko pracy systemu RAD C++. Struktura programu w RAD C++. Paleta komponentów. Komunikacja z użytkownikiem. Podstawowe instrukcje języka. Tworzenie kodu wynikowego.	W_01 U_02
3	Typy proste. Wybrane operatory, ich priorytet i łączność. Algorytmy przetwarzania iteracyjnego. Tablice i instrukcje pętli.	W_02 W_03 U_03
4-5	Typy wskaźnikowe. Operatory wskazania i wyłuskania. Arytmetyka adresowa. Praca z tablicami. Sterowanie w programie z wykorzystaniem instrukcji pętli.	W_02 W_03 U_03
6	Definiowanie funkcji i przekazywanie parametrów. Elementy programowania obiektowego w środowisku RAD C++. Właściwości i metody klas jako narzędzia działania na obiektach.	W_03 W_04 U_03 U_05
7	Klasa <i>AnsiString</i> – właściwości i metody. Przetwarzanie tekstów. Wykorzystanie komponentów <i>TStringGrid</i> , <i>TMemo</i> oraz klasy <i>TStrings</i> .	W_03 W_04 W_05
8	Organizacja dostępu do pliku tekstowego z wykorzystaniem komponentów dialogowych. Rola klasy <i>TStrings</i> w transmisji danych między aplikacją i plikiem tekstowym.	W_03 W_04 W_05 U_02
9	Rodzaje błędów w programie i możliwości ich diagnozowania. Testowanie programu. Obsługa wyjątków w programie z wykorzystaniem konstrukcji programistycznej <i>try ... catch</i> .	W_04 W_06 U_02 U_04
10-11	Obsługa struktur w środowisku formularza. Współpraca aplikacji z plikiem binarnym – zapis i odczyt z pliku.	W_02 W_03 U_03
12	Elementy grafiki.	W_04 W_05 U_04

2. Treści kształcenia w zakresie ćwiczeń

3. Treści kształcenia w zakresie zadań laboratoryjnych

Nr zajęć lab.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1-2	Struktura projektu w systemie RAD C++. Komunikacja aplikacji z użytkownikiem. Karta obiektów standardowych w RAD C++. Konwersja typów i operacje arytmetyczne. Sterowanie w programie.	W_01 W_02 W_03 U_02 K_02
3-5	Kolejność działań w programie. Algorytmy obliczeniowe. Przetwarzanie iteracyjne. Implementacja algorytmów obliczeniowych w aplikacji RAD C++. Tablice i instrukcje pętli: wyznaczanie statystyk z tablic numerycznych. Kolokwium nr 1.	W_02 W_03 U_01 U_02 U_03 K_02

6-7	Przetwarzanie iteracyjne tablic z wykorzystaniem klasy <i>TStringGrid</i> do wprowadzania danych i wyprowadzania wyników. Funkcja użytkownika jako składowa klasy i jako funkcja zewnętrzna. Zmienne globalne. Przekazywanie parametrów funkcji.	W_04 W_05 U_01 U_03 U_05 K_02
8-9	Przetwarzanie ciągów znaków – klasa <i>AnsiString</i> . Iteracyjne przetwarzanie tekstów: statystyki, wyszukiwanie i modyfikowanie tekstów. Sprawdzian nr 1.	W_04 W_05 U_01 U_03 U_05 K_02
10-12	Organizacja dostępu do plików tekstowych, komponenty dialogowe dostępu do plików dyskowych. Transmisja danych między pamięcią dyskową a pamięcią operacyjną. Tworzenie aplikacji współpracującej z plikiem tekstowym.	W_01 W_04 W_05 U_03 U_05 K_01 K_02
13	Diagnozowanie błędów w programie, testowanie programu. Obsługa wyjątków w programie z wykorzystaniem konstrukcji programistycznej <i>try ... catch</i> .	W_06 U_04
14-15	Kolokwium nr 2. Obsługa struktury. Współpraca aplikacji z plikami binarnymi. Analiza kodu źródłowego i jego rozbudowa.	W_01 W_04 W_05 U_03 U_04 U_05 K_01 K_02

4. Charakterystyka zadań projektowych

5. Charakterystyka zadań w ramach innych typów zajęć dydaktycznych

Metody sprawdzania efektów kształcenia

Symbol efektu	Metody sprawdzania efektów kształcenia (sposób sprawdzenia, w tym dla umiejętności – odwołanie do konkretnych zadań projektowych, laboratoryjnych, itp.)
W_01	Zadania do samodzielnego wykonania na zajęciach.
W_02	Sprawdzian nr 1 na zajęciach. Egzamin.
W_03	Sprawdzian nr 1 na zajęciach. Egzamin.
W_04	Zadania do samodzielnego wykonania na zajęciach. Egzamin.
W_05	Zadania do samodzielnego wykonania na zajęciach. Egzamin.
W_06	Zadania do samodzielnego wykonania na zajęciach. Egzamin.
U_01	Kolokwium nr 1 i nr 2 na zajęciach – opracowanie własnej aplikacji odpowiednio obliczeniowej i współpracującej z plikiem dyskowym. Praca przy komputerze.
U_02	Kolokwium nr 1 i nr 2 na zajęciach – opracowanie własnej aplikacji odpowiednio obliczeniowej i współpracującej z plikiem dyskowym. Praca przy komputerze.
U_03	Kolokwium nr 1 i nr 2 na zajęciach – opracowanie własnej aplikacji odpowiednio obliczeniowej i współpracującej z plikiem dyskowym. Praca przy komputerze.
U_04	Komentarze i dyskusja propozycji rozwiązań studenckich na ćwiczeniach i wykładach.
U_05	Kolokwium nr 2 na zajęciach – opracowanie własnej aplikacji współpracującej z tekstowym plikiem dyskowym. Praca przy komputerze.
K_01	Komentarze i dyskusja propozycji rozwiązań studenckich na ćwiczeniach i wykładach.
K_02	Obserwacja postawy studenta podczas zajęć laboratoryjnych oraz dyskusja problemów zgłoszonych przez studentów.

D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS			
Lp.	Rodzaj aktywności	Obciążenie studenta	Jednostka
1.	Udział w wykładach	24	h
2.	Udział w ćwiczeniach		
3.	Udział w laboratoriach	30	h
4.	Udział w zajęciach projektowych		
5.	Udział w konsultacjach (2-3 razy w semestrze)	6	h
6.	Konsultacje projektowe		
7.	Udział w egzaminie	3	h
8.			
9.	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	63	h
10.	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego <i>(1 punkt ECTS=27 godzin obciążenia studenta)</i>	2,3	ECTS
11.	Samodzielne studiowanie tematyki wykładów	22	h
12.	Samodzielne przygotowanie się do ćwiczeń		
13.	Samodzielne przygotowanie się do kolokwium	25	h
14.	Samodzielne przygotowanie się do laboratoriów	30	h
15.	Wykonanie sprawozdań		
16.	Przygotowanie do kolokwium końcowego z laboratorium		
17.	Wykonanie projektu lub dokumentacji		
18.	Przygotowanie do egzaminu	22	h
19.			
20.	Liczba godzin samodzielnej pracy studenta	99	h
21.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy <i>(1 punkt ECTS=27 godzin obciążenia studenta)</i>	3,7	ECTS
22.	Sumaryczne obciążenie pracą studenta	162	h
23.	Punkty ECTS za moduł <i>1 punkt ECTS=27 godzin obciążenia studenta</i>	6	ECTS
24.	Nakład pracy związany z zajęciami o charakterze praktycznym <i>Suma godzin związanych z zajęciami praktycznymi</i>	113	h
25.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym <i>1 punkt ECTS=27 godzin obciążenia studenta</i>	4,2	ECTS

E. LITERATURA

Wykaz literatury	<ol style="list-style-type: none">1. Daniluk A., <i>C++Builder Borland Developer Studio 2006. Kompendium programisty</i>, Helion, Gliwice 2006.2. Daniluk A., <i>C++ Builder 2006 ćwiczenia praktyczne</i>, Helion, Gliwice 2006.3. Dorobek M., <i>C++ Builder. Podręcznik, Seria: Biblioteka Programisty</i>, Wydawnictwo Mikom, Warszawa 2002.4. Kubiak M., <i>Programuję w językach C/C++ i C++ Builder</i>, Wydawnictwo Mikom, Warszawa 2003.5. Stasiewicz A., <i>Borland C++ całkiem inny świat</i>, Helion, Gliwice 1998.6. Stasiewicz A., <i>C++ Builder. 20 efektywnych programów</i>, Helion, Gliwice 2002.
Witryna WWW modułu/przedmiotu	