

Podstawy programowania komputerów		4,5 ECTS	
Computer Programming Basics		dzienne magisterskie Koszalin	
Kod przedmiotu	Imię i Nazwisko organizującego		
AOZI1000WSUPPRO	Walery Susłow		
LICZBA GODZIN	w	c	Kurs egzaminacyjny
	2	1	
Kursy poprzedzające	program szkoły ponadgimnazjalnej		
Słowa kluczowe	komputer, programowanie, język C		

PROGRAM SZCZEGÓŁOWY	
CELE KURSU.	
<i>Opis kursu – cele i procedury ich osiągnięcia:</i>	
<p>Na bazie wiedzy szkoły średniej uzyskanej w ramach przedmiotu „Technologie informatyczne” student nabywa wiedzę i umiejętności w zakresie programowania komputerów, które są konieczne do opanowania kolejnych kursów, wykorzystujących technologie informatyczne, oraz studiowania przedmiotów kierunkowych informatyki.</p>	
<i>Cel poznawczy:</i>	
<p>Poszerzenie wiedzy o technologii i językach programowania komputerów oraz dokładne opanowanie wiadomości dot. techniki programowania strukturalnego.</p>	
<i>Cel kształcący:</i>	
<p>Kształtowanie umiejętności samodzielnego planowania, pisania, testowania i konserwacji kodów aplikacji komputerowych w stylu strukturalnym w języku programowania ANSI C w stopniu pozwalającym przejść do dalszego studiowania współczesnych technologii programowania obiektowego.</p>	
<i>Cel praktyczny:</i>	
<p>Kształtowanie umiejętności przygotowania, kompilacji, debugowania i testowania kodów aplikacji strukturalnych w środowisku Borland C i C++ Builder.</p>	
<i>Umiejętności wynikowe kursu i ich ewaluacja:</i>	
<p>Wykonywanie podstawowych działań związanych z przygotowaniem algorytmu, napisaniem kodu i testowaniem aplikacji w języku ANSI C. Oceniane są umiejętności posługiwania się podstawowymi technikami programowania strukturalnego, w tym budowania aplikacji jako kompletu funkcji, wykorzystania własnych typów danych strukturalnych (tablic, struktur, unii), planowania obciążenia pamięci komputera.</p>	
<i>Wymagania podstawowe (stopnie dostateczne)</i>	<i>Wymagania rozszerzone (stopnie dobre)</i>
<p>Wykonuje poprawnie proste algorytmy liniowe w oparciu o podstawowe typy danych.</p> <p>Potrafi obsługiwać operacje wejścia/wyjścia, w tym potrafi programować akcje na konsoli oraz komunikować się z przestrzenią dyskową komputera za pomocą standardowych bibliotek ANSI C.</p> <p>Poprawnie kompiluje, uruchamia i debuguje nieduże aplikacje (do 100 linii kodu), potrafi analizować listingi tego rozmiaru.</p>	<p>Wykonuje poprawnie złożone algorytmy gałęziowe i cykliczne, potrafi wykorzystać dane niestandardowe, definiowane przez użytkownika.</p> <p>Używa niestandardowe biblioteki języka C, wykorzystuje rekursie, makrodefinicje, wskaźniki, zmienne rejestrowe, dynamiczną alokację pamięci.</p> <p>Potrafi optymalizować złożone algorytmy, pracuje samodzielnie z listingiem powyżej 200 linii kodu.</p>

Wykład 1. 2 godz.	OGÓLNE ZASADY PROGRAMOWANIA.
<p><i>Zagadnienia:</i> Proces powstawania oprogramowania. Potrzeby użytkownika oraz ich analiza, formułowanie zadania technicznego. Pojęcie algorytmu, podstawowe sposoby algorytmizacji zadań.</p> <p>Przegląd współczesnych języków programowania.</p>	
Wykład 2.	ELEMENTARZ ANSI C.
<p><i>Zagadnienia:</i> Elementy języka ANSI C: użytkowane symbole, identyfikatory, zastrzeżone słowa kluczowe. Elementy programu C (dyrektywy preprocesora, komentarze, deklaracje, instrukcje).</p> <p>Technologia tworzenia programu, testowanie i konserwacja, dokumentowanie programów. Pakiety programistyczne, charakterystyka pakietu Borland Builder. Pierwszy program w C.</p>	
Wykład 3.	DANE W ANSI C.
<p><i>Zagadnienia:</i> Dane: stałe i zmienne. Typy danych. Inicjalizacja danych. Deklaracja zmiennych. Wyświetlanie danych na ekranie. Standardowe funkcje i operatory do wprowadzania i wyprowadzania danych. Wzorce konwersji oraz sekwencje ucieczki. Przykładowe programy wykorzystujące wymianę danych z konsoli.</p>	
Wykład 4.	INSTRUKCJE W ANSI C.
<p><i>Zagadnienia:</i> Operatory i wyrażenia arytmetyczne i logiczne. Stałe symboliczne i makrodefinicje. Instrukcje: przypisania, sterujące, skoku, warunkowe, wyboru i pętli. Instrukcja złożona. Przykładowe programy wykorzystujące instrukcje złożone.</p>	
Wykład 5.	FUNKCJE I MAKROINSTRUKCJE W ANSI C.
<p><i>Zagadnienia:</i> Funkcje, nagłówek i ciało funkcji, funkcja <i>main</i>. Argumenty funkcji z wartościami domyślnymi. Zmienne lokalne, zewnętrzne, statyczne i rejestrowe. Zakres i widoczność zmiennej. Definiowanie zmiennych lokalnych, użycie stosu.</p> <p>Makroinstrukcje (makrodefinicje). Przykład programu z wykorzystaniem makrodefinicji.</p>	
Wykład 6.	TABLICE, WSKAŹNIKI I MACIERZE.
<p><i>Zagadnienia:</i> Wskaźniki, operatory adresowania i wyłuskiwania. Działania na wskaźnikach. Tablice liczbowe i znakowe. Użycie tablic w programie. Wskaźniki do tablic. Zakres tablicy. Przykładowe programy wykonujące działania na tablicach i wskaźnikach.</p>	
Wykład 7.	STRUKTURY I UNIE.
<p><i>Zagadnienia:</i> Struktury: inicjalizacja, działania na polach. Zmienne typu strukturalnego. Struktury zagnieżdżone. Dostęp do pól struktury, operacje przypisania. Struktura a funkcja, struktura a wskaźnik. Tablice i pliki struktur. Unie.</p> <p>Przykład wykorzystania struktury w prostej bazie danych.</p>	
Wykład 8.	REKURENCJA.
<p><i>Zagadnienia:</i> Definicja rekurencji. Przykład rekurencji w definicji funkcji silni. Szereg Fibonacciego. Problemy rekurencji. Rekurencja a grafika.</p> <p>Przykładowe programy do obliczeń rekurencyjnych.</p>	
Wykład 9.	OPERACJE NA PLIKACH W ANSI C.
<p><i>Zagadnienia:</i> Operacje na plikach: otwieranie, odczyt-zapis, zamykanie, dostęp niskopoziomowy (systemowy). Pojęcie strumienia danych. Buforowanie wejścia/wyjścia. Wskaźniki typu FILE. Otwieranie/zamykanie pliku. Odczyt i zapis plików z poziomu programu C.</p> <p>Przykład programu wykonującego operacje na plikach.</p>	
Wykład 10.	STEROWANIE PAMIĘCIĄ W C.

Zagadnienia: Pamięć na stosie i na sterwie. Pamięć a system operacyjny. Przydział pamięci w ruchu programu. Tablice statyczne i dynamiczne. Zwalnianie pamięci. Niepowodzenie w alokacji pamięci.

Przykład programu z wykorzystaniem „ręcznej” alokacji pamięci.

Wykład 11.	POLA BITOWE.
-------------------	---------------------

Zagadnienia: Sterowanie bitami. Dane typu „pola bitowe”. Ograniczenia pól bitowych. Operatory bitowe w C. Istotne możliwości bitowej reprezentacji danych.

Zaawansowana reprezentacja danych. Wybór reprezentacji danych. Tablica struktur. Tablica wskaźników. Listy łączone. Abstrakcyjne typy danych. Kolejka jako abstrakcyjny typ danych. Reprezentacja czasu w programie poprzez zdefiniowane typy. Biblioteka *time.h*.

Wykład 12.	ŁAŃCUCHY ZNAKOWE.
-------------------	--------------------------

Zagadnienia: Stała łańcuchowa. Inicjalizacja łańcucha. Wczytywanie łańcuchów. Wyświetlanie łańcuchów. Funkcje łańcuchowe. Badanie długości łańcuchów. Działania na łańcuchach.

Przykładowe programy wykonujące operacje na łańcuchach.

Wykład 13.	TYP WYLICZENIOWY. PROGRAMOWANIE NIESTRUKTURALNE.
-------------------	---

Zagadnienia: Wyliczenia. zmienne typu wyliczeniowego. Instrukcje na zmiennych *enum*. Określenie wartości stałych wyliczenia. Wprowadzenie i wyświetlenie łańcuchów za pomocą wyliczenia.

Instrukcje niestrukturalne: *break*, *continue*, *goto*, *return*, *switch*. Opcja „jump optimization” kompilatora. Przykład programu niestrukturalnego.

Wykład 14.	PROGRAMOWANIE WSPÓLBIEŻNE W C.
-------------------	---------------------------------------

Zagadnienia: Pseudo-współbieżne wykonywanie funkcji w C. Funkcje *setjmp* i *longjmp*. Przełączanie zadań. Sekwencje przełączające. Przykład programowania współbieżnego.

PLANY WYNIKOWE ĆWICZEŃ¹	
---	--

Ćw. 1. 2 godz.	ZAPOZNANIE SIĘ ZE ŚRODOWISKIEM PROGRAMOWYM.
-----------------------	--

Opis ćwiczenia – cele i procedury ich osiągania:

Kompilacja i konsolidacja prostego programu, dopasowanie środowiska programowego do potrzeb użytkownika, ustawienie i zapisanie konfiguracji, wyjaśnienie podstawowych konstrukcji programu ANSI C. Proste programy, wprowadzanie danych liczbowych, działania arytmetyczne na danych i wyprowadzanie wyników na ekran.

Umiejętności wynikowe i ich ewaluacja

Opanowanie podstawowych trybów pracy w środowisku programowym

Wymagania podstawowe

Samodzielne wprowadzenie kodu, kompilacja, uruchomienie i poprawienie błędów w bardzo prostych programach.

Wymagania rozszerzone (stopnie dobre)

Sprawne kojarzenie ustawień konfiguracyjnych środowiska. Samodzielna celowa zmiana kodu bardzo prostych programów.

Ćw. 2. 2 godz.	TYPY DANYCH JĘZYKA C.
-----------------------	------------------------------

Opis ćwiczenia – cele i procedury ich osiągania:

Podstawowe typy danych. Stałe liczbowe i tekstowe (*string*). Typy pochodne. Zakresy ważności i zasłanianie nazw. Modyfikator *const* i *#define*.

Umiejętności wynikowe i ich ewaluacja: Kształtowanie umiejętności deklaracji i wykorzystania danych w programie.

¹ Prowadzący zajęcia laboratoryjne są zobligowani do opracowania tematów o różnym stopniu trudności tak, by początkujący mogli zdobywać punkty ćwicząc rzeczy elementarne (dla nich wykonalne), a zaawansowani mogli zdobywać punkty poszerzając swoją wiedzę przez poznanie i praktyczne zastosowanie ciekawych algorytmów z innych pokrewnych dziedzin np. z metod numerycznych, badań operacyjnych, metod optymalizacji itp.

Wymagania podstawowe: Samodzielne dobieranie i deklaracja odpowiednich typów danych.	Wymagania rozszerzone (stopnie dobre): Optymalne wykorzystanie typów danych w kontekście programu.
Ćw. 3. 2 godz.	INSTRUKCJE PROSTE I ZŁOŻONE.
Opis ćwiczenia – cele i procedury ich osiągania: Instrukcja warunkowa <i>if</i> . Instrukcje pętli. Budowanie instrukcji złożonych z wykorzystaniem <i>switch</i> .	
Umiejętności wynikowe i ich ewaluacja: Opanowanie standardowych konstrukcji gałęziowych i cyklicznych w kodzie C.	
Wymagania podstawowe: Umiejętność zapisania kodu instrukcji złożonej w oparciu o gotowy algorytm.	Wymagania rozszerzone (stopnie dobre): Umiejętność zapisania instrukcji złożonej w kilku wariantach konstrukcji kodu.
Ćw. 4. 2 godz.	FUNKCJE PODSTAWOWYM MODULEM PROGRAMU C.
Opis ćwiczenia – cele i procedury ich osiągania: Programy zawierające funkcje. Zwracanie rezultatu przez funkcję. Funkcje inline (otwarte). Funkcje biblioteczne.	
Umiejętności wynikowe i ich ewaluacja: Potrafi napisać program składający się minimum z trzech funkcji.	
Wymagania podstawowe: Poprawnie deklaruje i definiuje funkcje użytkownika, potrafi zakodować wzajemne wywołanie funkcji.	Wymagania rozszerzone (stopnie dobre): Dokonuje optymalnego podziału zadania na moduły, które potrafi zakodować jako odrębne funkcje.
Ćw. 5. 2 godz.	DANE ROZBUDOWANE: TABLICE I WSKAŹNIKI.
Opis ćwiczenia – cele i procedury ich osiągania: Tablice - wprowadzenie. Tablice znakowe. Tablice wielowymiarowe. Wskaźniki - wstęp. Zastosowanie wskaźników w argumentach funkcji i wobec tablic.	
Umiejętności wynikowe i ich ewaluacja: Opanowanie techniki organizacji danych jednorodnych w postaci tablic i techniki odwołania się do danych poprzez wskaźniki.	
Wymagania podstawowe: Poprawnie deklaruje i wykorzystuje tablice i wskaźniki w przypadku danych prostych.	Wymagania rozszerzone (stopnie dobre) Optymalnie dobiera tablicową organizację danych do zadania, umie wykorzystać wskaźniki na danych złożonych.
Ćw. 6. 2 godz.	DANE ROZBUDOWANE: STRUKTURY I PLIKI.
Opis ćwiczenia – cele i procedury ich osiągania: Struktury. Pliki. Zapis struktury do pliku. Odczytanie struktury z pliku.	
Umiejętności wynikowe i ich ewaluacja: Opanowanie umiejętności obsługi plików i struktur w C.	
Wymagania podstawowe: Potrafi zakodować procedury obsługi plików w oparciu o standardowe funkcje biblioteczne. Umie zadeklarować i wykorzystać proste struktury.	Wymagania rozszerzone (stopnie dobre): Wybiera sposób obsługi plików odpowiedni do zadania, umie obsłużyć struktury złożone.
Ćw. 7. 2 godz.	INDYWIDUALNY PROJEKT ZALICZENIOWY.
Opis ćwiczenia – cele i procedury ich osiągania: Przykładowa tematyka zadań do wykonania z zakresu: 1. Operacje na typach liczbowych prostych: obliczanie pierwiastków równania; obliczanie punktów przecięcia dwóch krzywych, zadanych wzorem; obliczanie n pierwszych wyrazów ciągu harmonicznego; obliczanie silni oraz dwumianu Newtona; znajdowanie w ciągu liczb wprowadzonych przez użytkownika minimum, maksimum;	

<p>obliczanie wartości funkcji $y=f(x)$ dla kilku wartości x.</p> <p>2. Operacje na tablicach i ciągach znaków: posortowanie rosnąco ciągu liczb wprowadzonych przez użytkownika; określenie ilości wystąpień poszczególnych liter we wprowadzonym ciągu znaków; posortowanie malejąco liter we wprowadzonym ciągu znaków z uwzględnieniem kolejności polskich znaków w alfabecie; zastąpienie w ciągu znaków jednego wyrażenia przez inne z uwzględnieniem możliwości użycia i zastępowania znaków końca linii, tabulacji itp.</p> <p>3. Operacje z grafiką: rysowanie wykresu funkcji i nałożenia na niego aproksymacji; rysowanie prostej animacji; rozwiązanie zadania „Wieże Hanoi” dla n klocków;</p> <p>4. Program złożony z modułów: wykonywanie działań na macierzach; obliczenie średniej ważonej dla n podanych liczb, z odpowiadającymi im wagami; horoskop - podajemy datę urodzenia i imię, otrzymując znak zodiaku + znak horoskopu chińskiego + krótki opis osoby spod danego znaku.</p> <p>5. Operacje na złożonych strukturach danych: adresownik - lista nazwisk, imion, adresów, numerów telefonów osób z możliwością dodania, edycji, usuwania, porządkowania (np. po nazwisku, mieście, imionach), wyszukiwania, zapisu i odczytu z pliku; rozkład jazdy autobusów - lista miejscowości docelowych i tranzytowych wraz z godzinami odjazdu autobusów, z możliwością dodania, edycji, usuwania, porządkowania po miejscowości docelowej, godzinie przyjazdu, godzinie wyjazdu, oraz wyszukiwania, zapisu i odczytu z pliku.</p>	
<p><i>Umiejętności wynikowe i ich ewaluacja:</i> Umiejętność wykonania zakończonego projektu w postaci aplikacji komputerowej, która rozwiązuje konkretne zadanie i jest przygotowana w języku C z wykorzystaniem stylu programowania strukturalnego.</p>	
<p><i>Wymagania podstawowe:</i> Potrafi wytłumaczyć strukturę aplikacji oraz objaśnić działanie wszystkich fragmentów kodu we własnym projekcie.</p>	<p><i>Wymagania rozszerzone (stopnie dobre):</i> Umie wprowadzić na bieżąco zmiany do własnego projektu, które by poprawili zauważone przez prowadzącego mankamenty aplikacji.</p>

OPIS RYGORÓW POŚREDNICH	
Kolokwium I.	INSTRUKCJE PROSTE I ZŁOŻONE.
<p><i>Opis kolokwium :</i> 15-minutowa praca z wykorzystaniem przez studenta dowolnych materiałów własnych nad zapisaniem kodu w języku C na zadany temat.</p>	
<p><i>Sposób oceniania:</i> Analiza poprawności przedstawionego kodu</p>	
<p><i>Wymagania podstawowe:</i> Popełnione błędy łatwo mogą być poprawione podczas kompilacji.</p>	<p><i>Wymagania rozszerzone (stopnie dobre):</i> Kod dokładnie odpowiada zadaniu, wykorzystano nie standardowe, autorskie pomysły.</p>
Kolokwium II.	REPREZENTACJA DANYCH ROZBUDOWANYCH.
<p><i>Opis kolokwium :</i> 15-minutowa praca z wykorzystaniem przez studenta dowolnych materiałów własnych nad zapisaniem kodu w języku C na zadany temat.</p>	
<p><i>Sposób oceniania:</i> Analiza poprawności przedstawionego kodu</p>	
<p><i>Wymagania podstawowe:</i> Popełnione błędy łatwo mogą być poprawione podczas kompilacji.</p>	<p><i>Wymagania rozszerzone (stopnie dobre):</i> Kod dokładnie odpowiada zadaniu, wykorzystano nie standardowe, autorskie pomysły.</p>

Kolokwium III.		OPERACJE NA PLIKACH I STEROWANIE PAMIĘCIĄ.	
<i>Opis kolokwium</i> : 15-minutowa praca z wykorzystaniem przez studenta dowolnych materiałów własnych nad zapisaniem kodu w języku C na zadany temat.			
<i>Sposób oceniania</i> : Analiza poprawności przedstawionego kodu			
<i>Wymagania podstawowe</i> : Popołnione błędy łatwo mogą być poprawione podczas kompilacji.		<i>Wymagania rozszerzone (stopnie dobre)</i> : Kod dokładnie odpowiada zadaniu, wykorzystano nie standardowe, autorskie pomysły.	

OPIS RYGORU ZASADNICZEGO (ZALICZENIE).	
<i>Opis ogólny rygoru zasadniczego</i>	
Warunki zaliczenia: zaliczenie minimum dwóch kolokwium na oceną pozytywną oraz zaliczenie laboratorium na podstawie punktów za zajęcia, kartkówki oraz projekt.	
<i>Wymagania podstawowe</i>	<i>Wymagania rozszerzone (stopnie dobre)</i>

WYKAZ ŹRÓDEŁ (PODRĘCZNIKÓW I SKRYPTÓW)
<ol style="list-style-type: none"> 1. Brian W. Kernighan, Dennis M. Ritchie, Język ANSI C. – WNT, 2001. 2. Tony Zhang, Poznaj C w 24 godziny. – Wyd. Infoland, 2001. 3. Stephen Prata, Język C. Szkoła Programowania. – Wyd. Robomatic, Wrocław, 1999. 4. Marek Kotowski, Wysokie C. – Lupus, 1998. 5. Steven S. Skiena, Miguel A. Revilla, Wyzwania Programistyczne. – Wyd. Szkolne i Pedagogiczne, Warszawa, 2004.