

SYLABUS PRZEDMIOTU

1.	Nazwa przedmiotu w języku polskim oraz języku angielskim Praktyczny wstęp do programowania Practical introduction to programming
2.	Dyscyplina naukowa nauki fizyczne (1 ECTS) informatyka techniczna i telekomunikacja (1 ECTS) informatyka (2 ECTS)
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Wydział Fizyki i Astronomii
5.	Rodzaj przedmiotu (<i>obowiązkowy, do wyboru</i>) obowiązkowy
6.	Kierunek studiów Informatyka stosowana i systemy pomiarowe
7.	Poziom studiów I stopień
8.	Rok studiów I rok
9.	Semestr (<i>zimowy lub letni</i>) zimowy
10.	Forma zajęć i liczba godzin wykład – 15 godzin laboratorium komputerowe – 45 godzin
11.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu brak
12.	Cele kształcenia dla przedmiotu W ramach kursu studenci dowiedzą się na czym polega programowanie oraz jakie elementy składają się na program komputerowy. Nauczą się jak pisać, rozszerzać, kompilować, poprawiać i uruchamiać programy w języku C z elementami C++. Nauczą się jak rozwiązać podstawowe zagadnienia programistyczne w sposób praktyczny.

13.	Treści programowe Cel i motywacja programowania, języki programowania i środowiska programistyczne, tworzenie programu w języku C/C++ (struktura programu, podstawowe funkcje), kompilacja i uruchomienie programu. Instrukcje wyjścia, formatowanie wyjścia do konsoli w trybie ANSI. Pętle, instrukcje warunkowe if/switch. Prosta animacja w konsoli. Typy danych i operatory, wyrażenia warunkowe. Tablice danych 1D i 2D. Wskaźniki. Funkcje. Struktury danych. Procedury wejścia/wyjścia do plików. Biblioteka SFML. Format graficzny ppm i generowanie sekwencji plików do animacji. Elementy obiektowe języka C++. Przykłady praktyczne: programowanie prostych gier, algorytm generowania fraktali (np. Mandelbrota), kalkulator, operacje na danych liczbowych zapisanych w plikach: import, obróbka, eksport. Użycie bibliotek zewnętrznych rozszerzających standard języka.		
14.	<table border="1"> <tr> <td data-bbox="316 689 944 2038"> Zakładane efekty uczenia się Student: – zna miejsce i rolę języków C/C++ w informatyce; – zna podstawowe elementy konstrukcyjne języka C/C++ tj. pętle oraz instrukcje warunkowe; – zna budowę i sposób obsługi podstawowych struktur danych języka C/C++: tablic, struktur, wskaźników; – zna strukturę plików graficznych ppm (portable picture map); – zna przynajmniej jedną bibliotekę zewnętrzną języka C/C++ i sposób jej dołączania do projektu; – zna podstawy programowania w C++; – potrafi zapisać (z użyciem kodu) rysunek w postaci pliku ppm, procedurę tę wykorzystuje w praktyce; – potrafi napisać według własnego pomysłu program generujący prosty rysunek lub animację w konsoli (ANSI); – potrafi napisać program na podstawie zadanego algorytmu (np. fraktal Mandelbrota); – potrafi napisać prosty program do odczytu i zapisu danych do pliku; – wykorzystuje dokumentacje bibliotek zewnętrznych (np. SFML) podczas rozwiązywania postawionych zadań; – podczas realizacji zadań praktycznych samodzielnie wybiera odpowiednie struktury danych języka C/C++ oraz proponuje sposób rozwiązania postawionego zadania. </td><td data-bbox="944 689 1396 2038"> Symbole odpowiednich kierunkowych efektów uczenia się I1_W04 I1_U08 I1_K02, I1_K03 </td></tr> </table>	Zakładane efekty uczenia się Student: – zna miejsce i rolę języków C/C++ w informatyce; – zna podstawowe elementy konstrukcyjne języka C/C++ tj. pętle oraz instrukcje warunkowe; – zna budowę i sposób obsługi podstawowych struktur danych języka C/C++: tablic, struktur, wskaźników; – zna strukturę plików graficznych ppm (portable picture map); – zna przynajmniej jedną bibliotekę zewnętrzną języka C/C++ i sposób jej dołączania do projektu; – zna podstawy programowania w C++; – potrafi zapisać (z użyciem kodu) rysunek w postaci pliku ppm, procedurę tę wykorzystuje w praktyce; – potrafi napisać według własnego pomysłu program generujący prosty rysunek lub animację w konsoli (ANSI); – potrafi napisać program na podstawie zadanego algorytmu (np. fraktal Mandelbrota); – potrafi napisać prosty program do odczytu i zapisu danych do pliku; – wykorzystuje dokumentacje bibliotek zewnętrznych (np. SFML) podczas rozwiązywania postawionych zadań; – podczas realizacji zadań praktycznych samodzielnie wybiera odpowiednie struktury danych języka C/C++ oraz proponuje sposób rozwiązania postawionego zadania.	Symbole odpowiednich kierunkowych efektów uczenia się I1_W04 I1_U08 I1_K02, I1_K03
Zakładane efekty uczenia się Student: – zna miejsce i rolę języków C/C++ w informatyce; – zna podstawowe elementy konstrukcyjne języka C/C++ tj. pętle oraz instrukcje warunkowe; – zna budowę i sposób obsługi podstawowych struktur danych języka C/C++: tablic, struktur, wskaźników; – zna strukturę plików graficznych ppm (portable picture map); – zna przynajmniej jedną bibliotekę zewnętrzną języka C/C++ i sposób jej dołączania do projektu; – zna podstawy programowania w C++; – potrafi zapisać (z użyciem kodu) rysunek w postaci pliku ppm, procedurę tę wykorzystuje w praktyce; – potrafi napisać według własnego pomysłu program generujący prosty rysunek lub animację w konsoli (ANSI); – potrafi napisać program na podstawie zadanego algorytmu (np. fraktal Mandelbrota); – potrafi napisać prosty program do odczytu i zapisu danych do pliku; – wykorzystuje dokumentacje bibliotek zewnętrznych (np. SFML) podczas rozwiązywania postawionych zadań; – podczas realizacji zadań praktycznych samodzielnie wybiera odpowiednie struktury danych języka C/C++ oraz proponuje sposób rozwiązania postawionego zadania.	Symbole odpowiednich kierunkowych efektów uczenia się I1_W04 I1_U08 I1_K02, I1_K03		

15.	Literatura obowiązkowa i zalecana Literatura obowiązkowa: Brian W. Kernighan, Dennis M. Ritchie, 'Język ANSI C', 1987 Maciej Matyka, Kombinacje C++, Helion 2023 Literatura zalecana: Brian W. Kernighan, Dennis M. Ritchie, 'Język ANSI C. Programowanie. Wydanie II', Helion 2011 Grębosz J., Symfonia C++ standard: programowanie w języku C++, 2006 Zbigniew Koza, C++ w dwunastu długich krokach, 2006	
16.	Metody weryfikacji zakładanych efektów uczenia się: ciągła kontrola postępów w zakresie tematyki zajęć w formie werbalnych pytań prowadzącego oraz dyskusji rozwiązywanych zadań i problemów zawartych na listach zadań (w ramach ćwiczeń laboratoryjnych) krótkie kartkówki pisemne z wybranych zagadnień (w ramach ćwiczeń laboratoryjnych)	
17.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu: 0.8 oceny stanowi ocena z list zadań z ćwiczeń laboratoryjnych, 0.2 oceny ocena z kartkówek	
	Nakład pracy studenta wyrażony w godzinach zajęć oraz punktach ECTS	liczba godzin przeznaczona na zrealizowanie danego rodzaju zajęć
	zajęcia (wg planu studiów) z prowadzącym:	
	– wykład:	15
	– laboratorium:	45
	praca własna studenta (w tym udział w pracach grupowych):	
	– przygotowanie do zajęć:	15
	– czytanie wskazanej literatury:	10
	– przygotowanie programów:	15
	– przygotowanie do kartkówek:	5
	Łączna liczba godzin zajęć	105
	Liczba punktów ECTS (jeśli jest wymagana)	4