

SYLABUS PRZEDMIOTU

1.	Nazwa przedmiotu w języku polskim oraz języku angielskim Projekt C++ C++ project
2.	Dyscyplina naukowa informatyka techniczna i telekomunikacja
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 II rok
9.	Semestr (<i>zimowy lub letni</i>) zimowy
10.	Forma zajęć i liczba godzin laboratorium komputerowe – 30 godzin
11.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu wiedza i umiejętności z zakresu programowania w języku C++ na poziomie kursu Programowanie w C++
12.	Cele kształcenia dla przedmiotu Umiejętność samodzielnego zaplanowania i stworzenia dużej aplikacji w języku C++, wykorzystującej zaawansowane biblioteki graficzne lub systemowe, przy zastosowaniu sprawdzonych technik programistycznych i pracy z repozytorium. Umiejętność samodzielnego poznawania nowych bibliotek i interfejsów i narzędzi programistycznych potrzebnych w praktycznym rozwiązywaniu problemów pojawiających się w pracy nad projektem. Zdolność do samorozwoju. Umiejętność tworzenia specyfikacji technicznej zawierającej spis planowanych funkcjonalności projektu oraz dokumentacji programistycznej i instrukcji powstałej obsługi aplikacji.

13.	Treści programowe Tworzenie aplikacji z graficznym interfejsem użytkownika na dowolne urządzenie i system operacyjny. Programowanie aplikacji w języku C++ z wykorzystaniem bibliotek wysokiego poziomu, np. Qt, Coocos2D, SFML, openframeworks.cc, Cinder, itp. Praca w środowiskach wykorzystujących zaawansowane biblioteki i rozszerzenia standardowego języka.	
14.	Zakładane efekty uczenia się Student: – zna zasady programowania strukturalnego oraz obiektowego w języku C++, metody programowania graficznego interfejsu użytkownika, a także odpowiednie środowiska programistyczne (QtCreator, Visual Studio itp.) i narzędzia do testowania i dystrybucji oprogramowania; – potrafi posługiwać się systemem kontroli wersji git i aktywnie z niego korzysta podczas pracy nad projektem; – potrafi zastosować symulację zjawisk fizycznych, jeśli wymaga tego temat realizowanej aplikacji; – stosuje metody matematyczne i informatyczne do rozwiązywania problemów programistycznych podczas realizacji projektu; – zna zasady organizacji programów z graficznym interfejsem użytkownika (obsługa zdarzeń, zasoby, layouty, akcje, itd.). Korzysta z bibliotek rozszerzających standard języka, np. Qt, Windows forms, SFML, OpenGL, DirectX. Potrafi napisać w C++ aplikację z graficznym interfejsem użytkownika lub aplikację mobilną; – umie zaplanować funkcjonalności i cele projektu oraz realistyczny harmonogram pracy i go przestrzegać; – stosuje techniki programowania "zwinnego" (agile); – potrafi samodzielnie zdobywać wiedzę i umiejętności konieczne do wykonania projektu; – potrafi stworzyć wstępną specyfikację oraz końcową dokumentację techniczną projektu; – potrafi zaprezentować zagadnienie techniczne dotyczące tematyki realizowanego projektu.	Symbole odpowiednich kierunkowych efektów uczenia się I1_W04, I1_W05 I1_U03, I1_U04, I1_U06, I1_U08, I1_U11 I1_K04

15.	Literatura obowiązkowa: https://git-scm.com/docs/gittutorial Literatura zalecana: https://en.wikipedia.org/wiki/Agile_software_development https://terminalroot.com/meet-15-alternatives-to-create-graphical-interfaces-with-cpp/ https://en.cppreference.com/w/cpp/links/libs	
16.	Metody weryfikacji zakładanych efektów uczenia się: pisemna praca semestralna (opis projektu) przygotowanie wystąpienia ustnego (prezentacja projektu) przygotowanie i zrealizowanie projektu programistycznego systematyczne konsultacje w ciągu semestru kontrola przebiegu pracy poprzez analizę historii repozytorium git	
17.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu: ciągła kontrola obecności i postępów w zakresie tematyki zajęć wystąpienie ustne (prezentacja założeń projektu) przygotowanie i zrealizowanie projektu indywidualnego napisanie raportu z zajęć	
	Nakład pracy studenta wyrażony w godzinach zajęć oraz punktach ECTS	liczba godzin przeznaczona na zrealizowanie danego rodzaju zajęć
	zajęcia (wg planu studiów) z prowadzącym: – laboratorium:	30
	praca własna studenta (w tym udział w pracach grupowych): – przygotowanie do zajęć: – czytanie wskazanej literatury: – praca nad realizacją projektu: – napisanie raportu z projektu:	5 10 75 10
	Łączna liczba godzin zajęć	130
	Liczba punktów ECTS (<i>jeśli jest wymagana</i>)	5