

## OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim <b>Projekt C++/C++ project</b>
2.	Dyscyplina <b>Informatyka techniczna i telekomunikacja - 5</b>
3.	Język wykładowy <b>polski</b>
4.	Jednostka prowadząca przedmiot <b>Wydział Fizyki i Astronomii</b>
5.	Kod przedmiotu/modułu <b>24-ISSP-S1-E3-PC++</b>
6.	Rodzaj przedmiotu/modułu ( <i>obowiązkowy lub do wyboru</i> ) <b>Obowiązkowy</b>
7.	Kierunek studiów (specjalność/specjalizacja) <b>Informatyka stosowana i systemy pomiarowe</b>
8.	Poziom studiów ( <i>I lub II stopień, jednolite studia magisterskie, studia doktoranckie</i> ) <b>I stopień</b>
9.	Rok studiów ( <i>jeśli obowiązuje</i> ) 2
10.	Semestr ( <i>zimowy lub letni</i> ) <b>zimowy</b>
11.	Forma zajęć i liczba godzin <b>Laboratorium 30 godz</b> Metody nauczania dyskusja w grupie, ćwiczenia laboratoryjne
12.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia <b>dr Tomasz Golan</b>
13.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <b>Zaliczony kurs „Programowanie w C++”</b>
14.	Cele przedmiotu <b>Celem zajęć jest wykształcenie praktycznych umiejętności programowania aplikacji w języku C++ z wykorzystaniem bibliotek wysokiego poziomu, np. Qt, Coocos2D, SFML. Zajęcia rozszerzają i ugruntowują wiedzę i umiejętności zdobyte na kursie „Programowanie w C++”. Dają one nową perspektywę na sposób tworzenia programów komputerowych i zmuszają do pracy w środowiskach opartych o różne biblioteki i rozszerzenia standardowego języka. W ramach zajęć studenci tworzą aplikację z</b>

	graficznym interfejsem użytkownika na dowolne urządzenie i system operacyjny.	
15.	Treści programowe <b>Zaawansowana biblioteka oparta na C++, np. Qt (alternatywy: Cocos2D, SFML, openframeworks.cc, Cinder i inne)</b>	
16.	<p>Zakładane efekty uczenia się</p> <p>Po ukończeniu kursu student:</p> <ul style="list-style-type: none"> <li>• <b>zna:</b> <ul style="list-style-type: none"> <li>○ Sposób programowania w środowisku wysokiego poziomu opartym na C++</li> <li>○ Sposób posługiwania się zintegrowanym środowiskiem programistycznym (np. QtCreator) i/lub bibliotekami rozszerzającymi standard języka</li> <li>○ Zasady organizacji programów z graficznym interfejsem użytkownika (obsługa zdarzeń, zasoby, definicja GUI, layouts, actions, etc.)</li> <li>○ Podstawowe mechanizmy wybranego systemu kontroli wersji (np. git lub mercurial)</li> </ul> </li> <li>• <b>potrafi:</b> <ul style="list-style-type: none"> <li>○ Napisać w C++ aplikację “okienkową” posiadającą graficzny interfejs użytkownika</li> <li>○ Napisać w C++ aplikację (np. grę) na urządzenie wbudowane</li> <li>○ Posługiwać się rozproszonym systemem kontroli wersji (np. git lub mercurial) i publicznymi repozytoriami (np. github, bitbucket)</li> <li>○ Zaplanować poprawnie cele projektu</li> <li>○ Zaplanować harmonogram projektu</li> <li>○ Zaprezentować zagadnienie techniczne dotyczące tematyki związanej ze swoim projektem</li> <li>○ Osiągnąć zaplanowane cele projektu</li> </ul> </li> <li>• <b>dostrzega:</b> <ul style="list-style-type: none"> <li>○ związki pomiędzy różnymi technologiami komputerowymi (np. QTQuick, QML i Java Script, definiowaniem właściwości obiektów w Qt i CSS)</li> <li>○ potrzebę wykonania prototypu we wczesnej fazie projektu</li> <li>○ trudności związane z poprawnym rozplanowaniem zadań i celów pośrednich w projektach programistycznych</li> </ul> </li> </ul>	<p>Symbole odpowiednich kierunkowych efektów uczenia się:</p> <p><b>I1_W04, I1_W05, I1_U03, I1_U04, I1_U06, I1_U08, I1_U11, I1_K04</b></p>
17.	<p>Literatura obowiązkowa i zalecana (<i>źródła, opracowania, podręczniki itp.</i>)</p> <ol style="list-style-type: none"> <li>1. <a href="http://qt-project.org/doc/">http://qt-project.org/doc/</a></li> <li>2. <b>C++ i Qt. Wprowadzenie do wzorców projektowych. Wydanie II, Helion 2014</b></li> <li>3. <b>Źródła internetowe (dokumentacje bibliotek wymienionych w treściach</b></li> </ol>	

	<b>programowych, przykłady i podręczniki).</b>
18.	Metody weryfikacji zakładanych efektów uczenia się: - przygotowanie i zrealizowanie projektu (indywidualnego lub grupowego) – ocena poszczególnych komponentów
19.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu:</p> <p>W ramach zajęć studenci zdobywają punkty za konkretne aktywności związane z projektem. Aktywności podzielone są na 5 kategorii.</p> <p>1- plan projektu (na GitHub lub BitBucket) - max <b>15 pkt</b></p> <ul style="list-style-type: none"> <li>• opis końcowego projektu, opis funkcjonalności, co chcę zrobić, jak, czym</li> <li>• harmonogram pracy (co najmniej 3 tzw. kamienie milowe z datami włączając prezentację prototypu 5 minut)</li> <li>• data oddania końcowego projektu (deadline)</li> </ul> <p>tzw. projekty trywialne, np. kółko i krzyżyk, proste aplikacje QT itp. do napisania w kilka wieczorów będą odrzucane na etapie planu projektu do poprawki.</p> <p>2- prezentacja techniczna (15 minut) - 10 pkt (można mieć <b>max 20 pkt</b>)</p> <ul style="list-style-type: none"> <li>• tematy dotyczące technologii, algorytmów, przeglądów oprogramowania itp. Przykładowe tematy: git i wersjonowanie, graficzne nakładki, doxygen, zaawansowane C++/C++-x11, GUI, grafika, algorytmy itp.itd (również swoje własne tematy). Maksymalnie 2 na zajęcia. Maksymalnie 2 na studenta. Tylko do końca roku 2016.</li> <li>• ocena: czas, jakość, stopień zaawansowania, użyteczność dla innych studentów</li> </ul> <p>3- prototyp i prezentacja prototypu (dla wszystkich, 5 minut) - <b>15 pkt</b></p> <ul style="list-style-type: none"> <li>• czas prezentacji (5 minut)</li> <li>• co założyłem</li> <li>• co zrobiłem</li> <li>• opis/pokaz</li> <li>• jakie problemy były</li> <li>• co planuję zrobić</li> <li>• jakie problemy przewiduję</li> </ul> <p>4- końcowy projekt - <b>35 pkt</b></p> <ul style="list-style-type: none"> <li>• zgodność z założeniami</li> <li>• termin</li> </ul>

	<ul style="list-style-type: none"> <li>• kamienie milowe</li> <li>• systematyczność i jakość projektu w repozytorium</li> <li>• jakość kodu</li> <li>• użyte technologie i stopień zaawansowania projektu (port androida, dodatkowe biblioteki, użyte algorytmy, grafika komputerowa, użycie bazy danych, front www)</li> </ul> <p>5- prezentacja projektu - <b>15 pkt</b></p> <ul style="list-style-type: none"> <li>• multimedia, prezentacja - oceniane przez innego prowadzącego</li> </ul> <p>Na zajęciach dopuszczamy możliwość pracy w parach, ale na każdym etapie (plan projektu, prezentacje itd.) musi być wyszczególniony i pokazany wkład każdego członka zespołu. Również stopień zaawansowania projektu powinien być uwzględniony i dostosowany do ilości osób. Osoby pracujące razem punkty zdobywają osobno.</p> <p>Proponowana skala ocen:</p> <p>od 90 do 100 pkt - 5.0  od 80 do 89 pkt - 4.5  od 70 do 79 pkt - 4.0  od 60 do 69 pkt - 3.5  od 51 do 59 pkt - 3.0  do 50 pkt - 2.0</p>	
20.	20. Nakład pracy studenta/doktoranta	
	forma działań studenta/doktoranta	liczba godzin na realizację działań
	Zajęcia (wg planu studiów) z prowadzącym: - wykład: - konwersatorium: - laboratorium: - inne:	- - 30 -
	Praca własna studenta, doktoranta (w tym udział w pracach grupowych) np.: - przygotowanie do zajęć: - opracowanie wyników: - czytanie wskazanej literatury: -przygotowanie prac/wystąpień/projektów: - napisanie raportu z zajęć: - przygotowanie do sprawdzianów i egzaminu:	90 - - 10 - -
	Łączna liczba godzin	130
	Liczba punktów ECTS	5

