

OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim Projekt C++	
2.	Nazwa przedmiotu/modułu w języku angielskim C++ project	
3.	Jednostka prowadząca przedmiot WFiA	
4.	Kod przedmiotu/modułu 24-ISSP-S1-E3-PC++	
5.	Rodzaj przedmiotu/modułu (<i>obowiązkowy lub fakultatywny</i>) Obowiązkowy	
6.	Kierunek studiów Informatyka stosowana i systemy pomiarowe	
7.	Poziom studiów (<i>I lub II stopień lub jednolite studia magisterskie</i>) I stopień	
8.	Rok studiów (<i>jeśli obowiązuje</i>) 2	
9.	Semestr (<i>zimowy lub letni</i>) 3 (zimowy)	
10.	Forma zajęć i liczba godzin Laboratorium 30 godz.	
11.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia dr Maciej Matyka, dr Tomasz Golan, mgr Jarosław Gołembiewski	
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu oraz zrealizowanych przedmiotów Zaliczone kursy „Programowanie w C++”	
13.	Cele przedmiotu 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 graficznym interfejsem użytkownika na dowolne urządzenie i system operacyjny.	
14.	Zakładane efekty kształcenia Po ukończeniu kursu student: • zna: o Sposób programowania w środowisku wysokiego	Symbole kierunkowych efektów kształcenia, np.: <i>K_W01*</i> , <i>K_U05</i> , <i>K_K03</i>

	<p>poziomu opartym na C++</p> <ul style="list-style-type: none"> o Sposób posługiwania się zintegrowanym środowiskiem programistycznym (np. QtCreator) i/lub bibliotekami rozszerzającymi standard języka o Zasady organizacji programów z graficznym interfejsem użytkownika (obsługa zdarzeń, zasoby, definicja GUI, layouts, actions, etc.) o Podstawowe mechanizmy wybranego systemu kontroli wersji (np. git lub mercurial) • potrafi: <ul style="list-style-type: none"> o Napisać w C++ aplikację “okienkową” posiadającą graficzny interfejs użytkownika o Napisać w C++ aplikację (np. grę) na urządzenie wbudowane o Posługiwać się rozproszonym systemem kontroli wersji (np. git lub mercurial) i publicznymi repozytoriami (np. github, bitbucket) o Zaplanować poprawnie cele projektu o Zaplanować harmonogram projektu o Zaprezentować zagadnienie techniczne dotyczące tematyki związanej ze swoim projektem o Osiągnąć zaplanowane cele projektu • dostrzega: <ul style="list-style-type: none"> o związki pomiędzy różnymi technologiami komputerowymi (np. QTQuick, QML i Java Script, definiowaniem właściwości obiektów w Qt i CSS) o potrzebę wykonania prototypu we wczesnej fazie projektu o trudności związane z poprawnym rozplanowaniem zadań i celów pośrednich w projektach programistycznych 	<p>K_W04, K_W05, K_U03, K_U04, K_U06, K_U08, K_U11, K_K04</p>
15.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Zaawansowana biblioteka oparta na C++, np. Qt (alternatywy: Cocos2D, SFML, openframeworks.cc, Cinder i inne) 	
16.	<p>Zalecana literatura (<i>podręczniki</i>)</p> <ol style="list-style-type: none"> 1. http://qt-project.org/doc/ 2. C++ i Qt. Wprowadzenie do wzorców projektowych. Wydanie II, Helion 2014 3. Źródła internetowe (dokumentacje bibliotek wymienionych w treściach programowych, przykłady i podręczniki). 	
17.	<p>Forma zaliczenia poszczególnych komponentów przedmiotu/modułu, sposób sprawdzenia osiągnięcia zamierzonych efektów kształcenia:</p> <p>wykład: seminarium: laboratorium: konwersatorium: inne:</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 15 pkt</p> <ul style="list-style-type: none"> • opis końcowego projektu, opis funkcjonalności, co chcę zrobić, jak, czym • harmonogram pracy (co najmniej 3 tzw. kamienie milowe z datami włączając 	

prezentację prototypu 5 minut)

- data oddania końcowego projektu (deadline)

tzw. projekty trywialne, np. kółko i krzyżyk, proste aplikacje QT itp. do napisania w kilka wieczorów będą odrzucane na etapie planu projektu do poprawki.

2- prezentacja techniczna (15 minut) - 10 pkt (można mieć **max 20 pkt**)

- 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.
- ocena: czas, jakość, stopień zaawansowania, użyteczność dla innych studentów

3- prototyp i prezentacja prototypu (dla wszystkich, 5 minut) - **15 pkt**

- czas prezentacji (5 minut)
- co założyłem
- co zrobiłem
- opis/pokaz
- jakie problemy były
- co planuję zrobić
- jakie problemy przewiduję

4- końcowy projekt - **35 pkt**

- zgodność z założeniami
- termin
- kamienie milowe
- systematyczność i jakość projektu w repozytorium
- jakość kodu
- użyte technologie i stopień zaawansowania projektu (port androida, dodatkowe biblioteki, użyte algorytmy, grafika komputerowa, użycie bazy danych, front www)

5- prezentacja projektu - **15 pkt**

- multimedia, prezentacja - oceniane przez innego prowadzącego

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.

	Proponowana skala ocen: 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	
18.	Język wykładowy Polski	
19.	Obciążenie pracą studenta	
	Forma aktywności studenta	Średnia liczba godzin na zrealizowanie aktywności
	Godziny zajęć (wg planu studiów) z nauczycielem: - wykład: - ćwiczenia: - laboratorium: - inne:	30
	Praca własna studenta np.: - przygotowanie do zajęć i praca w domu: - opracowanie wyników: - czytanie wskazanej literatury: - napisanie raportu z zajęć: - przygotowanie do egzaminu:	90 – – 10 –
	Suma godzin	130
	Liczba punktów ECTS	5

*objaśnienie symboli:

K (przed podkreśleniem) - kierunkowe efekty kształcenia
 W - kategoria wiedzy
 U - kategoria umiejętności
 K (po podkreśleniu) - kategoria kompetencji społecznych
 01, 02, 03 i kolejne - numer efektu kształcenia