

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Systemy i aplikacje bez granic (ubiquitous)</b>		Kod <b>1010511361010500089</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>3 / 6</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stoień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>30</b> Ćwiczenia: <b>-</b> Laboratoria: <b>30</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>4</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b> <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>4 100%</b> <b>4 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b> dr inż. Bartłomiej Prędko email: Bartlomiej.Predki@cs.put.poznan.pl tel. (61) 665-2932 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu funkcjonowania komputera i programowania imperatywnego (zdobytą na zajęciach z przedmiotów Wprowadzenie do informatyki i Podstawy programowania) oraz wybranych elementów sieci komputerowych.
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu implementacji i oceny kosztu działania prostych algorytmów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	<b>Kompetencje społeczne</b>	Powinien również rozumieć konieczność poszerzania swoich kompetencji i wykazywać gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
1. Przekazanie studentom podstawowej wiedzy na temat historii i funkcjonowania systemów powszechnych i mobilnych.		
2. Pozyskanie przez studentów umiejętności projektowania i programowania systemów powszechnych, programowania i przetwarzania danych w chmurze.		
3. Przekazanie wiedzy dotyczącej programowania komunikacji bezprzewodowej pomiędzy różnymi kategoriami urządzeń.		
4. Kształtowanie u studentów umiejętności pracy zespołowej w trakcie realizacji projektu na zajęciach laboratoryjnych.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie tzw. systemów i aplikacji bez granic - [K1st_W4]		
2. ma wiedzę o istotnych kierunkach rozwoju i najważniejszych osiągnięciach informatyki w zakresie tzw. systemów i aplikacji bez granic - [K1st_W5]		
3. zna podstawowe techniki, metody oraz narzędzia wykorzystywane w procesie rozwiązywania zadań informatycznych w zakresie tzw. systemów i aplikacji bez granic - [K1st_W7]		
4. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych, systemów operacyjnych - [-]		
<b>Umiejętności:</b>		

1. pozyskiwać informacje z zakresu tzw. systemów i aplikacji bez granic z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie - [K1st\_U1]
2. potrafi odpowiednio posługiwać się technikami informacyjno-komunikacyjnymi, znajdującymi zastosowanie na różnych etapach realizacji przedsięwzięć informatycznych z zakresu tzw. systemów i aplikacji bez granic - [K1st\_U2]
3. potrafi, formułując i rozwiązując zadania informatyczne z zakresu tzw. systemów i aplikacji bez granic, zastosować odpowiednio dobrane metody, w tym eksperymentalne - [K1st\_U4]
4. potrafi zaprojektować system informatyczny z zakresu tzw. systemów i aplikacji bez granic, dobierając język programowania do danego zadania programistycznego z zakresu tzw. systemów i aplikacji bez granic oraz używając właściwych metod, technik i narzędzi - [K1st\_U10]
5. ma umiejętność formułowania algorytmów i ich implementacji z użyciem przynajmniej jednego z popularnych narzędzi stosowanych w tzw. systemach i aplikacjach bez granic - [K1st\_U11]
6. potrafi planować i realizować proces własnego permanentnego uczenia się oraz zna możliwości dalszego doksztalcania się w zakresie tzw. systemów i aplikacji bez granic - [K1st\_U19]

#### Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności z zakresu tzw. systemów i aplikacji bez granic bardzo szybko stają się przestarzałe - [K1st\_K1]
2. ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich z zakresu tzw. systemów i aplikacji bez granic oraz zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych - [K1st\_K2]

### Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów:
  - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach;
- b) w zakresie laboratoriów:
  - na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
- ocenianie ciągle, na każdych zajęciach (odpowiedzi ustne) - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,
- ocenę wiedzy i umiejętności wykazanych na teście pisemnym w formie kilkunastu pytań, do zaliczenia testu potrzebne jest minimum 50% poprawnych odpowiedzi.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- wykazanie się ciekawymi umiejętnościami ponadprogramowymi,
- omówienia dodatkowych aspektów zagadnienia,
- przygotowanie opracowania na określony,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
- uwagi związane z udoskonaleniem materiałów dydaktycznych,

### Treści programowe

W ramach wykładu przedstawiane są następujące zagadnienia:

- programowanie na platformie iOS, a w szczególności:
- podstawy języków programowania Swift i Objective-C,
- programowanie z wykorzystaniem różnorodnych API dla poszczególnych platform sprzętowych - iPhone, iPad, Apple Watch,
- programowanie z wykorzystaniem platform uniwersalnych na przykładzie Xamarin,
- korzystanie z usług chmurowych, np. Dropbox, Google Drive, iCloud, OneDrive z poziomu API aplikacji,
- metodologia komunikacji urządzeń - sieci bezprzewodowe, Bluetooth,
- protokoły wymiany danych, np. JSON, REST,
- wstęp do chmur obliczeniowych, np. Microsoft Azure,
- wykorzystanie nietypowych urządzeń, np. beaconów, itp.

W ramach ćwiczeń laboratoryjnych studenci będą mogli w praktyce przećwiczyć materiały prezentowane na wykładach w formie mini-projektów obejmujących od jednego do kilku zajęć laboratoryjnych.

Metody dydaktyczne:

1. Wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, dyskusja i analiza problemów.		
2. Ćwiczenia laboratoryjne: wykonywanie eksperymentów, rozwiązywanie zadań, dyskusja, praca w zespole.		
<b>Literatura podstawowa:</b>		
1. iOS 5: programowanie: receptury / Vandad Nahavandipoor ; [tł.: Robert Górczyński], Helion 2013.		
2. Tworzenie aplikacji na platformę iOS 5 : z wykorzystaniem Xcode, Interface Builder, Instruments, GDB oraz innych kluczowych narzędzi, Brandon Alexander, J. Bradford Dillon, Kevin Y. Kim, Helion, 2012		
3. Objective-C : praktyczny podręcznik tworzenia aplikacji na systemy iOS i Mac OS X!, Stephen G. Kochan, Helion 2012		
4. Podstawy języka Swift: programowanie aplikacji dla platformy iOS / Mark A. Lassoﬀ &#38;#38; Tom Stachowitz, Helion 2016		
5. Service design patterns: fundamental design solutions for SOAP/WSDL and RESTful Web services, Robert Daigneau, Addison-Wesley, 2012		
6. Inteligentny dom: automatyzacja mieszkania za pomocą platformy Arduino, systemu Android i zwykłego komputera / Mike Riley, Helion 2013		
7. Android : programowanie aplikacji / Dawn Griffiths, David Griffiths, Helion 2016		
<b>Literatura uzupełniająca:</b>		
1. The Swift Programming Language 3.1, Apple Inc., 2017		
2. Using Swift with Cocoa and Objective-C, Apple Inc., 2014		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>	<b>Czas (godz.)</b>	
1. udział w zajęciach laboratoryjnych	30	
2. przygotowanie do ćwiczeń laboratoryjnych:	5	
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych:	5	
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2 8	
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	5	
6. przygotowanie do sprawdzianów / kolokwium	30	
7. udział w wykładach	15	
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 150 stron		
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2
Zajęcia o charakterze praktycznym	53	2