

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Systemy i aplikacje bez granic (ubiquitous)		Kod 1010511361010510089
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 3 / 6
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 4 100%
Odpowiedzialny za przedmiot / wykładowca:		
<p>dr inż. Bartłomiej Prędko email: Bartlomiej.Predki@cs.put.poznan.pl tel. (61) 665-2932 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań</p>		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	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	Umiejętności:	Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu implementacji i oceny kosztu działania prostych algorytmów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	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.
Cel przedmiotu:		
<ol style="list-style-type: none"> Przekazanie studentom podstawowej wiedzy na temat historii i funkcjonowania systemów powszechnych i mobilnych. Pozyskanie przez studentów umiejętności projektowania i programowania systemów powszechnych, programowania i przetwarzania danych w chmurze. Przekazanie wiedzy dotyczącej programowania komunikacji bezprzewodowej pomiędzy różnymi kategoriami urządzeń. Kształtowanie u studentów umiejętności pracy zespołowej w trakcie realizacji projektu na zajęciach laboratoryjnych. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
<ol style="list-style-type: none"> ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych, systemów operacyjnych - [K_W4] ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych - [K_W6] zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu budowy systemów komputerowych, systemów operacyjnych - [K_W8] 		
Umiejętności:		

1. pozyskiwać informacje 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 - [K_U1]
2. posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych - [K_U6]
3. zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi - [K_U21]
4. ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z popularnych narzędzi - [K_U22]

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]

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 każdym 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,
- wskazywanie trudności percepcyjnych studentów, umożliwiające bieżące doskonalenia procesu dydaktycznego.

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. Microsoft Kinect, beaconów, itp.

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

Literatura podstawowa:		
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. The Swift Programming Language 3.1, Apple Inc., 2017		
5. Using Swift with Cocoa and Objective-C, Apple Inc., 2014		
6. Podstawy języka Swift : programowanie aplikacji dla platformy iOS / Mark A. Lassoﬀ & Tom Stachowitz, Helion 2016		
7. Service design patterns : fundamental design solutions for SOAP/WSDL and RESTful Web services, Robert Daigneau, Addison-Wesley, 2012		
8. Inteligentny dom : automatyzacja mieszkania za pomocą platformy Arduino, systemu Android i zwykłego komputera / Mike Riley, Helion 2013		
9. Android : programowanie aplikacji / Dawn Griffiths, David Griffiths, Helion 2016		
Literatura uzupełniająca:		
Bilans nakładu pracy przeciętnego studenta		
Czynność		Czas (godz.)
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		3
		8
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)		4
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		
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	63	2
Zajęcia o charakterze praktycznym	58	2