

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu <b>Projektowanie i konstrukcja systemów rozproszonych 1</b>		Kod <b>1010515311010511656</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 1</b>
Ścieżka obieralności/specjalność <b>Sieci komputerowe</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>16</b> Ćwiczenia: - Laboratoria: - Projekty/seminaria: <b>16</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. Cezary Sobaniec email: cezary.sobaniec@put.poznan.pl tel. 61 665 2370 Informatyki ul. Piotrowo 2, 60-965 Poznań		
<b>Odpowiedzialny za przedmiot / wykładowca:</b> mgr inż. Jan Kończak email: jan.konczak@put.poznan.pl tel. 61 665 2942 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 podstawową wiedzę w zakresie: systemów operacyjnych, technologii sieciowych, przetwarzania rozproszonego, bezpieczeństwa systemów informatycznych oraz baz danych.
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność pozyskiwania informacji z literatury oraz innych źródeł (w języku ojczystym i angielskim); powinien potrafić wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne; powinien potrafić integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne.
3	<b>Kompetencje społeczne</b>	Student powinien również rozumieć konieczność poszerzania swoich kompetencji. 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 z zakresu konstrukcji nowoczesnych i skalowalnych systemów rozproszonych w architekturze zorientowanej na usługi (SOA) z wykorzystaniem usług sieciowych Web Services i REST. 2. Przekazanie studentom podstawowej wiedzy z zakresu projektowania i konstrukcji usług sieciowych pracujących w modelu asynchronicznym oraz z zakresu asynchronicznej komunikacji. 3. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z projektowaniem i implementacją usług sieciowych.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. ma zaawansowaną i pogłębioną wiedzę z zakresu szeroko rozumianych aplikacji i usług webowych, podstaw teoretycznych ich budowania oraz metod, narzędzi i środowisk programistycznych wykorzystywanych do ich implementacji - [K2st_W1] 2. ma zaawansowaną wiedzę szczegółową dotyczącą projektowania i implementacji usług sieciowych w modelu REST, asynchronicznej komunikacji z wykorzystaniem protokołu HTTP oraz implementacji usług sieciowych w modelu asynchronicznym - [K2st_W3] 3. ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia systemów informatycznych - [K2st_W5] 4. zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich i prowadzeniu prac badawczych w wybranym obszarze informatyki - [K2st_W6]		
<b>Umiejętności:</b>		

<ol style="list-style-type: none"><li>1. potrafi ? przy formułowaniu i rozwiązywaniu zadań inżynierskich ? integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5]</li><li>2. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st_U6]</li><li>3. potrafi poprawnie użyć wybraną metodę szacowania pracochłonności wytwarzania oprogramowania - [K2st_U7]</li><li>4. potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych oraz zaproponować ich ulepszenia (usprawnienia) - [K2st_U8]</li><li>5. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi - [K2st_U9]</li><li>6. potrafi ? zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne ? zaprojektować złożone urządzenie, system informatyczny lub proces oraz zrealizować ten projekt ? co najmniej w części ? używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K2st_U11]</li><li>7. potrafi współdziałać w zespole, przyjmując w nim różne role - [K2st_U15]</li></ol>
<b>Kompetencje społeczne:</b>
<ol style="list-style-type: none"><li>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]</li><li>2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych - [K2st_K2]</li></ol>

<b>Sposoby sprawdzenia efektów kształcenia</b>
<p>Wykład zaliczany jest pisemnym sprawdzianem o charakterze problemowym (5 pytań otwartych, każde dające 1 pkt., zaliczenie od 50%).</p> <p>Zaliczenie laboratorium wymaga realizacji dwóch projektów programistycznych. Ocenie podlega funkcjonalność, poprawność, zgodność z wymaganiami i rozumienie użytych technologii.</p>
<b>Treści programowe</b>
<p>Program wykładu obejmuje następujące zagadnienia:</p> <ol style="list-style-type: none"><li>1. Technologie HTML5: ewolucja języków znacznikowych, standard XML, aplikacje języka XML, XML Information Set, prezentacja dokumentów XML, standard XSLT i XSL Formatting Objects, XHTML, geneza HTML5.</li><li>2. Standard XML, aplikacje języka XML (MathML, DocBook, SVG, Open Document Format, Office Open XML), XML Information Set, prezentacja dokumentów XML, standard Extensible Stylesheet Language, XSL Formatting Objects, XHTML</li><li>3. Protokół HTTP: żądania i odpowiedzi, uwierzytelnianie, buforowanie podręczne, HTTP/2.</li><li>4. Asynchroniczna komunikacja z serwerem: AJAX, WebSocket.</li><li>5. Technologie dynamiczne po stronie serwera: CGI, FastCGI, moduły wbudowane w serwer, SSI, PHP, ASP, Java EE, .Net.</li><li>6. Architektura SOA: definicja usługi, motywacja dla SOA, definicja architektury, założenia SOA, magistrala ESB, usługi sieciowe Web Services: przegląd standardów WS-*, protokół SOAP, format komunikatów SOAP, wiązanie SOAP z protokołami transportowymi, standard opisu usług WSDL, profile WS-I (Basic, Security).</li><li>7. Usługi sieciowe REST: problem adresacji usług sieciowych, styl architektoniczny REST, zasoby i ich reprezentacje, metody i kody błędów protokołu HTTP, projektowanie usług.</li><li>8. Architektura zorientowana na zasoby (ROA): modele usług sieciowych, znaczenie adresów URI, hipermedia w REST, modelowanie REST, problem wyboru reprezentacji zasobów, stan interakcji w usługach REST, granularność zasobów, zasoby specjalne, kolekcje zasobów, powiązania między zasobami, mikroformaty, serwery buforujące, walidatory aktualizacji, problem idempotentności operacji POST.</li></ol> <p>Zajęcia laboratoryjne prowadzone są w formie 2-godzinnych ćwiczeń, odbywających się w laboratorium. Ćwiczenia realizowane są indywidualnie lub w zespołach 2 osobowych w zależności od charakteru ćwiczeń. Program laboratorium obejmuje następujące zagadnienia:</p> <ol style="list-style-type: none"><li>1. Warstwa prezentacyjna: HTML5, CSS3, JavaScript.</li><li>2. Protokół HTTP: analiza i generowanie żądań i odpowiedzi, uwierzytelnianie, przekierowania, negocjacja treści, ciasteczka, serwery pośredniczące.</li><li>3. Skrypty CGI: konfiguracja, tworzenie dynamicznych odpowiedzi, dostęp do baz danych, języki Perl, Python, budowa prostej aplikacji.</li><li>4. Usługi sieciowe REST: modelowanie usług sieciowych REST.</li><li>5. Usługi sieciowe REST od strony klienta.</li><li>6. Implementacja usług sieciowych REST.</li><li>7. XML: poprawność formatowania i poprawność dokumentów, DTD, projekt aplikacji języka XML, prezentacja dokumentów XML</li><li>8. XSLT: transformacje dokumentów do HTML offline i w przeglądarce, łączenie ze stylem kaskadowym, tworzenie odnośników</li></ol>

<b>Literatura podstawowa:</b>		
<b>Literatura uzupełniająca:</b>		
1. Dokumenty RFC specyfikujące protokoły webowe (m.in. HTTP, WebSocket).		
2. Python Tutorial, dostępny online: <a href="https://docs.python.org/3/tutorial/">https://docs.python.org/3/tutorial/</a>		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>	<b>Czas (godz.)</b>	
1. udział w zajęciach laboratoryjnych:	16	
2. przygotowanie do ćwiczeń laboratoryjnych:	12	
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2	
4. udział w wykładach	16	
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	30	
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron	10	
7. przygotowanie do kolokwium i obecność na kolokwium: 12 godz. + 1 godz.	13	
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	99	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	35	1
Zajęcia o charakterze praktycznym	52	2