

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Technologie programistyczne - systemy internetowe		Kod 1010512311010510639
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 1
Ścieżka obieralności/specjalność Technologie wytwarzania oprogramowania	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: -		Liczba punktów 5
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 %) 5 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Tomasz Pawlak email: Tomasz.Pawlak@cs.put.poznan.pl tel. 61 6653022 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K1st_W4, K1st_W5, K1st_W6, K1st_W7, weryfikowane w procesie rekrutacji na studia II stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
2	Umiejętności:	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K1st_U1, K1st_U2, K1st_U4, K1st_U5, K1st_U6, K1st_U7, K1st_U8, K1st_U9, K1st_U10, K1st_U11, K1st_U12, K1st_U14, K1st_U15, K1st_U19, weryfikowane w procesie rekrutacji na studia II stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
3	Kompetencje społeczne	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K1st_K1, K1st_K2, K1st_K3, K1st_K4, weryfikowane w procesie rekrutacji na studia II stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl 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: 1. Przekazanie studentom podstawowej wiedzy dotyczącej wykorzystania narzędzi programistycznych, w zakresie wytwarzania systemów internetowych. 2. Rozwijanie u studentów umiejętności rozwiązywania problemów dotyczących analizy i projektowania systemów internetowych w ramach technologii stosowanych w biznesie. 3. Omówienie zagadnień związanych z bezpieczeństwem i wydajnością systemów internetowych stosowanych w biznesie. 4. Kształtowanie u studentów umiejętności pracy zespołowej w ramach ćwiczeń laboratoryjnych w trakcie wykonywania projektów technicznych w grupach.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych i technologii służących do projektowania i implementacji systemów internetowych - [K2st_W1] 2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: zaprojektowanie, wytworzenie i testowanie systemu internetowego. - [K2st_W3] 3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce w zakresie - [K2st_W4] 4. ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych lub programowych - [K2st_W5] 5. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki - [K2st_W6]		
Umiejętności:		

1. potrafi pozyskiwać informacje z literatury, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski - [K2st_U1] 2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K2st_U16] 3. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5] 4. potrafi formułować i testować hipotezy związane z problemami inżynierskimi - [K2st_U4] 5. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st_U6] 6. potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych oraz zaproponować ich ulepszenia (usprawnienia) - [K2st_U8] 7. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi - [K2st_U9] 8. potrafi wybrać język programowania odpowiedni do danego zadania programistycznego - [K2st_U11] 9. potrafi - zgodnie z zadaną specyfikacją, uwzględniając aspekty pozatechniczne - zaprojektować system informatyczny - co najmniej w części - używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K2st_U11] 10. potrafi porozumiewać się w języku polskim i angielskim przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach, także z wykorzystaniem narzędzi informatycznych - [K2st_U12]
Kompetencje społeczne:
1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1] 2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i projektowych - [K2st_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 wykładach, b) w zakresie laboratoriów: - na podstawie oceny bieżącego postępu realizacji zadań laboratoryjnych, Ocena podsumowująca: a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez: - ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym w formie testu wielokrotnego wyboru, Egzamin składa się ze: - Zbioru od 8 do 12 pytań zamkniętych, z których na każde pytanie można udzielić jedną prawidłową odpowiedź z czterech możliwych. Za każdą poprawną odpowiedź uzyskuje się 1 punkt, a za błędną odejmowana jest 1/3 punktu. - Zbioru od 3 do 5 pytań otwartych, za które można uzyskać od 2 do 4 punktów. Aby uzyskać ocenę 3,0 należy uzyskać minimum 51% punktów, 3,5 - 61%, 4,0 - 71%, 4,5 - 81%, 5,0 - 91%. Czas odpowiedzi na pytania 1h. - omówienie wyników egzaminu, b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez: - ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych obejmujących programowanie, - ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole. Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za: - omówienia dodatkowych aspektów zagadnienia, - 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, - prezentacje autoreferatu powiązanego z tematem zajęć.
Treści programowe

Program wykładu obejmuje następujące zagadnienia:

- Komunikacja w systemach internetowych. Wykład wprowadza do zagadnień związanych z protokołami HTTP, XML-RPC oraz usługami Web-services. Wykład wprowadza do protokołu XML-RPC służącego do wymiany danych przy pomocy ramek XML. Następna część wykładu poświęcona jest wprowadzeniu do technologii Web Services korzystającej z protokołu HTTP. Omówiony zostanie także wzorzec usług sieciowych REST.

- Technologie wspierane przez przeglądarkę: HTML, CSS. Omówiona zostanie funkcjonalność języka oraz mniej popularne funkcje wpływające na ergonomię korzystania z aplikacji internetowej. Przedstawione zostaną rozszerzenia dostępne w HTML 5 i CSS 3. Przedstawienie preprocesorów CSS: Sass i Less.

- Technologie wspierane przez przeglądarkę: JavaScript, DOM, XSLT(XPath). Przedstawione zostaną zaawansowane metody wykorzystania języka np. budowanie wyrażeń lambda, pseudoobiektowość oraz narzędzia wspomagające programowanie w języku JavaScript. Przedstawiony zostanie model DOM dokumentów HTML i XML oraz związan z nim API. Omówiony zostanie język transformacji XSLT w kontekście tworzenia fragmentów stron internetowych z plików XML.

- Mechanizm Ajax i JSON. Omówione zostanie działanie mechanizmu dostępu do danych pochodzących z innych witryn (Cross-origin resource sharing). Na wykładzie zostanie dokładnie omówiony mechanizm Ajax. Omówione zostaną nowe rozwiązania służące do komunikacji z serwerem wprowadzone w ramach projektu HTML 5. Następnie przedstawiony zostanie format JSON przysyłania danych oraz biblioteka jQuery ułatwiająca manipulację drzewem DOM oraz wykonywanie zapytań AJAX.

- Dynamiczne generowanie stron i zasobów WWW: Servlet'y i JSP. Wykład wprowadza do mechanizmów dynamicznego generowania stron WWW oraz struktury aplikacji internetowej w technologii JEE. Omówiona zostanie technologia Servlet znajdująca się w kontenerach WWW w technologii JEE. Następnie pokazane zostanie zastosowanie szablonów JSP do generowania stron HTML i innych udostępnianych zasobów.

- Trwałość danych w aplikacji internetowej na przykładzie JDBC i JPA. Wykład przedstawia sposoby przechowywania i pobierania danych przy pomocy Java Database Connectivity (JDBC). Omówiony zostanie interfejs JPA służący do utrwalania obiektów Java w bazie danych. Omówienie relacyjnych baz danych oraz baz typu NoSQL na przykładzie systemów MongoDB, Redis oraz Apache Cassandra.

- Wzorzec MVC. Przegląd środowisk zgodnych z wzorcem MVC i jego pochodnymi.

- Tworzenie aplikacji w Ruby on Rails i Sinatra - wykład wprowadza do języka Ruby i przedstawia środowiska Rails i Sinatra służące do tworzenia aplikacji internetowych.

- Tworzenie aplikacji internetowych w środowisku .NET - wykład przedstawia środowisko ASP.NET (MVC) w zakresie możliwości wytwarzania aplikacji internetowych.

- Różne sposoby tworzenia aplikacji WWW na przykładach języka Dart i stosu MEAN: MongoDB + Express + AngularJS + Node.js. Dwa wykłady obejmują omówienie popularnych środowisk programistycznych pozwalających na tworzenie aplikacji zgodnych z modelem MVC. Środowiska prezentują różne paradygmaty stosowane w programowaniu aplikacji internetowych.

- Bezpieczeństwo, wydajność i poprawność aplikacji internetowych. Dwa wykłady wprowadzają w zagadnienia związane z tworzeniem bezpiecznych i wydajnych aplikacji internetowych. Omówione zostaną podstawowe zagrożenia, jakie można napotkać podczas eksploatacji aplikacji internetowych. Przedstawione zostaną metody pomiaru wydajności aplikacji internetowej oraz wpływające na nią czynniki.

Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych sesją instruktażową na początku semestru. Ćwiczenia realizowane są samodzielnie lub przez 2-osobowe zespoły studentów. Ćwiczenia realizowane są na podstawie materiałów dydaktycznych dostarczonych przez prowadzącego. Program laboratorium obejmuje następujące zagadnienia:

Realizacja ćwiczeń wprowadzających do technologii HTTP - 2 godz.

Rozwiązywanie zadań wprowadzających do technologii HTML5, CSS - 4 godz.

Rozwiązywanie zadań wprowadzających do technologii DOM, XSLT, JavaScript - 4 godz.

Realizacja ćwiczeń wprowadzających do technologii Servlet, JSP - 4 godz.

Omówienie i ćwiczenia z technologiami JDBC, JPA i bazami NoSQL - 4 godz.

Prezentacja technologii Ruby on Rails i realizacja prostych ćwiczeń - 4 godz.

Omówienie i ćwiczenia z technologiami Dart, Express, AngularJS, Node.js- 8 godz.

Metody dydaktyczne:

1. wykład: prezentacja multimedialna.

2. ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, praca w zespole, demonstracja.

Literatura podstawowa:

1. Krzysztof Rychlicki-Kicior: Java EE 6 : programowanie aplikacji www, Helion, 2010

2. Steven Sanderson, Adam Freeman: ASP.NET MVC 3 framework : zaawansowane programowanie, Helion, 2012

3. Semmy Perwal: Learning Web App Development, O'Reilly Media, 2014

4. Ethan Brown: Web Development with Node and Express, O'Reilly Media, 2014

Literatura uzupełniająca:

1. Dokumentacja standardów technicznych WWW: <http://www.w3.org>

2. Dokumentacja bibliotek Rails i Dart: <http://rubyonrails.org/>, <https://www.dartlang.org/>

3. Standardy związane z technologią J2EE: <http://www.jcp.org>

Bilans nakładu pracy przeciętnego studenta		
Czynność		Czas (godz.)
1. udział w zajęciach laboratoryjnych		30
2. przygotowanie do ćwiczeń laboratoryjnych		15
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych		15
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną)		6
5. udział w wykładach		30
6. napisanie programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)		10
7. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi		7
8. omówienie wyników egzaminu		2
9. przygotowanie do egzaminu		8
10. przeprowadzenie egzaminu		2
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	125	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	70	3
Zajęcia o charakterze praktycznym	60	3