

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Internet Applications</b>		Kod <b>1010514351010519080</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>3 / 5</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>angielski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>16</b> Ćwiczenia: <b>-</b> Laboratoria: <b>16</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>		Podział ECTS (liczba i %) <b>4 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
<p>dr inż. Marek Wojciechowski            email: Marek.Wojciechowski@cs.put.poznan.pl            tel. (0-61) 665-2962            Instytut Informatyki            ul. Piotrowo 2, 60-965 Poznań</p>		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu protokołów sieciowych, systemów baz danych i programowania zorientowanego obiektowo.
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność programowania prostych aplikacji z wykorzystaniem zintegrowanych środowisk programistycznych.
3	<b>Kompetencje społeczne</b>	Powinien również rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu.
<b>Cel przedmiotu:</b>		
<p>1. Uzyskanie wiedzy dotyczącej podstawowych pojęć i koncepcji z zakresu implementacji dokumentów WWW i aplikacji WWW, niezbędnej do rozróżniania podstawowych architektur aplikacji internetowych oraz metod implementacji ich modułów.</p> <p>2. Uporządkowanie wiedzy z zakresu architektur sieciowych, protokołów sieciowych, bezpieczeństwa systemów rozproszonych.</p> <p>3. Nabycie umiejętności implementacji aplikacji WWW z wykorzystaniem podstawowych technologii implementacji interfejsu użytkownika, m.in. HTML, CSS, JavaScript, Java Applets, XML+XSL, technologii implementacji logiki prezentacji, m.in. CGI, serwlety Java, JavaServer Pages, PHP, Active Server Pages, technologii implementacji logiki biznesowej, m.in. JavaBeans, biblioteki znaczników JSP.</p> <p>4. Pozyskanie kompetencji społecznych dotyczących zespołowego przygotowania projektu, w tym organizacji pracy zespołowej, a zwłaszcza przywództwa oraz komunikatywności w procesie grupowego rozwiązywania problemu.</p>		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
<p>1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektur aplikacji WWW wykorzystujących bazy danych - [K_W4,K_W5]</p> <p>2. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu projektowania, implementacji i wdrażania aplikacji WWW - [K_W8]</p> <p>3. ma uporządkowaną wiedzę z zakresu protokołów sieciowych i bezpieczeństwa systemów rozproszonych - [K_W4]</p>		
<b>Umiejętności:</b>		

1. potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi - [K_U21]
2. ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z popularnych narzędzi - [K_U22]
3. ma umiejętność tworzenia prostych aplikacji internetowych, wykorzystujących systemy baz danych oraz interakcyjne interfejsy użytkownika - [K_U24]
4. potrafi zaprojektować dobry interfejs użytkownika dla różnych klas systemów informatycznych - [K_U25]
5. potrafi wybrać technologię implementacji aplikacji WWW odpowiednią do zadanego obszaru zastosowań - [K_U20]
<b>Kompetencje społeczne:</b>
1. potrafi uczestniczyć w zespołowym przygotowaniu projektu, w tym zaplanować organizację pracy zespołowej oraz komunikować się w procesie grupowego rozwiązywania problemu - [K_K5,K_K6]

<b>Sposoby sprawdzenia efektów kształcenia</b>
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 ćwiczeń: - na podstawie oceny bieżącego postępu realizacji zadań, Ocena podsumowująca: Sprawdzanie założonych efektów kształcenia realizowane jest przez: - ocenę i obronę przez studenta sprawozdania z realizacji projektu, - ocenę wiedzy i umiejętności wykazanych na zaliczeniu pisemnym o formie testu jednokrotnego wyboru składającego się z 15 pytań, łączna liczba punktów za prawidłowe odpowiedzi: 15, minimalna liczba punktów umożliwiających zaliczenie: 8 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, - wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenie procesu dydaktycznego.
<b>Treści programowe</b>

<p>1. Wprowadzenie do architektury WWW i problematyki aplikacji WWW: historia rozwoju technologii WWW, składniki podstawowej architektury WWW: klient HTTP, serwer HTTP, protokół http, pojęcia aplikacji WWW, serwera aplikacji oraz komponentowe modele aplikacji WWW, podstawowe własności języka HTML.</p> <p>2. Interfejs użytkownika aplikacji WWW: arkusze stylów CSS i ich wykorzystanie do formatowania dokumentów HTML; język XML z uwzględnieniem ogólnej struktury dokumentu, opisu struktury za pomocą DTD i przestrzeni nazw, język XHTML, będący wersją HTML dostosowaną do reguł języka XML; język XSL służący do transformowania i formatowania dokumentów XML. Omówione zostaną zarówno część XSL dotycząca transformacji dokumentów (XSLT), jak i część dotycząca opisu sposobu formatowania (XSL-FO); znaczniki języka HTML służące do konstruowania formularzy; język skryptowy JavaScript, a w szczególności podstawy języka (deklaracje zmiennych, wyrażenia, operatory języka, struktury kontrolne i instrukcje, w tym pętle i instrukcje warunkowe) oraz model DOM (ang. Document Object Model) definiujący interakcję między skryptem JavaScript i obiektami zawartymi w dokumencie HTML; metody walidacji danych wprowadzanych przez użytkownika do formularzy HTML oraz metody manipulowania zawartością okna i dokumentu; omówienie metod pracy z łańcuchami znaków i datami oraz biblioteką oferującą bogaty zestaw funkcji matematycznych; prezentacja możliwości tworzenia interfejsu użytkownika za pomocą apletów języka Java.</p> <p>3. Przegląd własności funkcjonalnych serwerów HTTP na przykładzie serwera Apache: wewnętrzna architektura serwera Apache, jego podstawowe parametry konfiguracyjne, strukturę i zastosowania dziennika serwera, mechanizmy odwzorowania logicznych ścieżek dostępu w ścieżki fizyczne, dyrektywy blokowe oraz funkcje ochrony dostępu do dokumentów znajdujących się po stronie serwera Apache.</p> <p>4. Technologie implementacji logiki prezentacji: klasyfikacja metod konstrukcji logiki prezentacji oraz omówienie dwóch przykładowych technologii implementacji serwetów: technologii CGI i technologii serwetów Java; architektura i sposoby implementacji programów CGI, architektura serwetów Java, sposoby implementacji serwetów Java, obsługa nagłówków HTTP, zmiennych Cookies i sesji HttpSession; najważniejsze technologie szablonów (server pages) wykorzystywanych w aplikacjach WWW; starszych technologii Server Side Includes (SSI) oraz Active Server Pages (ASP), a następnie bardziej szczegółowo ? następcy ASP, czyli ASP.NET, oraz PHP; metody tworzenia logiki prezentacji aplikacji internetowej przy wykorzystaniu technologii szablonów rozwijanych dla języka Java; podstawowa technologia szablonów dla języka Java - technologia JSP (ang. Java Server Pages); cykl życia aplikacji JSP oraz podstawowe składowe technologii: deklaracje, dyrektywy i skryptety; język EL (ang. Expression Language), który ułatwia tworzenie złożonych aplikacji JSP; komplementarne technologie szablonów dla języka Java: Velocity, WebMacro i FreeMarker.</p> <p>5. Zagadnienia tworzenia logiki biznesowej przy wykorzystaniu komponentów JavaBean i bibliotek znaczników JSP: koncepcja komponentów JavaBean wraz z ich specyfikacją; zagadnienia związane z wykorzystaniem komponentów JavaBean na stronach JSP; wzorce projektowe wykorzystujące komponenty JavaBean do komunikacji między poszczególnymi warstwami aplikacji internetowej; technologia bibliotek znaczników JSP ? podstawowemu narzędziu służącemu do tworzenia dużych i złożonych aplikacji JSP; koncepcja i implementacja mechanizmu bibliotek znaczników na podstawie JSTL (ang. Java Standard Tag Library), standardowej biblioteki znaczników; przykłady ilustrujące sposób wykorzystania kilku rodzajów znaczników, m.in. znaczników podstawowych, formatujących, przetwarzających XML oraz obsługujących połączenie z bazą danych; metoda tworzenia własnych znaczników JSP.</p> <p>6. Mechanizmy dostępu do baz danych w najpopularniejszych technologiach do tworzenia aplikacji WWW: Java EE, ASP.NET i PHP. W kontekście technologii Java EE przedstawione będą podstawy JDBC, charakterystyczny dla aplikacji Java EE mechanizm uzyskiwania połączeń z bazą danych poprzez źródła danych oraz wprowadzenie do technologii odwzorowania obiektowo-relacyjnego (O/RM) i standardu Java Persistence.</p> <p>7. Funkcjonalność wymagana w aplikacjach internetowych, powtarzająca się niezależnie od konkretnych zastosowań i określana mianem infrastruktury. Po ogólnym wprowadzeniu przedstawiona będzie architektura Web Forms dla ASP.NET oraz architektura Model-View-Controller i jej popularna implementacja dla platformy Java EE ? Struts. Następnie omówiona zostanie technologia JavaServer Faces (JSF), ułatwiająca tworzenie aplikacji Java EE poprzez dostarczenie infrastruktury dla stanowego, komponentowego interfejsu użytkownika.</p> <p>8. Przegląd własności funkcjonalnych serwerów HTTP na przykładzie serwera Apache. W ramach wykładu omówimy wewnętrzną architekturę serwera Apache, jego podstawowe parametry konfiguracyjne, strukturę i zastosowania dziennika serwera, mechanizmy odwzorowania logicznych ścieżek dostępu w ścieżki fizyczne, dyrektywy blokowe oraz funkcje ochrony dostępu do dokumentów znajdujących się po stronie serwera Apache.</p> <p>9. Najważniejsze metody atakowania aplikacji WWW oraz mechanizmy ochrony przed nimi. Poruszone zostaną następujące rodzaje zagrożeń: kradzież kodu źródłowego aplikacji JSP, atak na pola ukryte HTML, atak na zmienne Cookies, atak typu Path Traversal, atak typu SQL Injection, przejęcie sesji, atak typu Cross-Site Scripting i atak typu Denial of Service. Część wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.</p> <p>Metody dydaktyczne:</p> <ol style="list-style-type: none"><li>wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań, demonstracja.</li><li>ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, praca w zespole, warsztaty.</li></ol>
<p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"><li>Praca zbiorowa, CSS i Ajax. Strony www zgodne ze standardami sieciowymi W3C, Helion, 2008.</li><li>A. Marciniak, JavaServer Faces i Eclipse Galileo. Tworzenie aplikacji www, Helion, 2010.</li><li>L. Bruce, S. Remy, Wprowadzenie do HTML 5. Autorytety informatyki, Helion, 2011.</li><li>L. Jesse, H. Dan, M. Brian, ASP.NET 2.0 i Ajax. Wprowadzenie, O'Reilly, 2008.</li></ol>
<p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"><li>P. Kazienko, K. Gwiazda, XML na poważnie, Helion, 2002.</li><li>L. Jesse, H. Dan, M. Brian, ASP.NET 2.0 i Ajax. Wprowadzenie, O'Reilly, 2008.</li><li>E. Jendrock, I. Evans, D. Gollapudi, K. Haase, C. Srivathsa, The Java EE 6 Tutorial, Oracle, 2010.</li></ol>

<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>	<b>Czas (godz.)</b>	
1. udział w zajęciach laboratoryjnych / ćwiczeniach:	16	
2. przygotowanie do ćwiczeń laboratoryjnych:	16	
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2	
4. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	18	
5. przygotowanie do sprawdzianów / kolokwium	12	
6. udział w wykładach	16	
7. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym	16	
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	96	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1
Zajęcia o charakterze praktycznym	50	2