

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Programowanie aplikacji internetowych</b>		Kod <b>1010515321010510505</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 2</b>
Ścieżka obieralności/specjalność <b>Aplikacje mobilne i wbudowane dla</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>12</b> Ćwiczenia: - Laboratoria: <b>20</b> Projekty/seminaria: -		Liczba punktów <b>3</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>3 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
<p>dr hab. inż. Rafał Różycki            email: rrozycy@put.poznan.pl            tel. 61 6653025            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 na temat sposobu funkcjonowania sieci Internet, znać podstawowe zasady protokołu HTTP oraz język HTML.
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność tworzenia statycznych witryn WWW, podstawową umiejętność programowania w dowolnym języku imperatywnym, znajomość podstaw systemów baz danych oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	<b>Kompetencje społeczne</b>	Powinien również rozumieć konieczność poszerzania swoich kompetencji i mieć 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>		
<ol style="list-style-type: none"> <li>Przekazanie studentom podstawowej wiedzy na temat: architektur aplikacji internetowych, wybranych technologii tworzenia aplikacji internetowych.</li> <li>Rozwijanie u studentów umiejętności projektowania i tworzenia aplikacji internetowych w wybranej technologii</li> <li>Kształtowanie u studentów umiejętności pracy zespołowej.</li> </ol>		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
<ol style="list-style-type: none"> <li>ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie technologii internetowych, wybranych języków i paradygmatów programowania, - [K_W4]</li> <li>ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: client site processing, server site processing. - [K_W5]</li> <li>ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w zakresie technologii internetowych - [K_W6]</li> <li>ma podstawową wiedzę o cyklu życia aplikacji internetowej - [K_W7]</li> <li>zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z zakresu projektowania aplikacji internetowych - [K_W8]</li> </ol>		
<b>Umiejętności:</b>		

1. potrafi 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. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K\_U5]
3. 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 - [K\_U10]
4. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K\_U13]
5. potrafi zaproponować ulepszenia (usprawnienia) istniejących aplikacji internetowych - [K\_U21]
6. potrafi ocenić przydatność metod i narzędzi projektowania aplikacji internetowych, w tym dostrzec ograniczenia tych metod i narzędzi - [K\_U24]
7. potrafi - stosując m.in. koncepcyjnie nowe metody - rozwiązywać złożone zadania informatyczne, w tym zadania nietypowe oraz zadania zawierające komponent badawczy - [K\_U25]
8. potrafi - zgodnie z zadaną specyfikacją, uwzględniając aspekty pozatechniczne - zaprojektować aplikację internetową oraz zrealizować ten projekt ? co najmniej w części ? używając właściwych metod, technik i narzędzi. - [K\_U27]

#### Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K\_K1]
2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet - [K\_K4]
3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K\_K6]

#### Sposoby sprawdzenia efektów kształcenia

Ocena formująca:

- a) w zakresie wykładów:
  - na podstawie przygotowanych krótkich prezentacji omawiających wybrane zagadnienia,
- b) w zakresie laboratoriów:
  - na podstawie oceny realizacji zadań cząstkowych,

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
    - ocenę przedstawionych prezentacji i/lub wiedzy oraz umiejętności wykazanych na zaliczeniu pisemnym w formie testu wielokrotnego wyboru, test składa się z kilkunastu pytań (max 15). Dla każdego pytania przygotowane są cztery odpowiedzi, z których każda może być poprawna lub błędna. Test jest zaliczony, gdy student zdobędzie przynajmniej połowę punktów
    - omówienie wyników zaliczenia
  - b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
    - ocenianie ciągle, na każdym zajęciach (odpowiedzi ustne) ? premiowanie przyrostu umiejętności posługiwania się poznаныmi zasadami i metodami,
    - ocenę rozbudowywanej w trakcie zajęć witryny WWW,
    - ocenę i obronę przez studenta sprawozdania z realizacji projektu (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,
  - wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

#### Treści programowe

<p>Program wykładu obejmuje następujące zagadnienia:</p> <p>Styczne witryny internetowe ? wybrane elementy języka HTML 5 (nowe elementy HTML w porównaniu z wersjami poprzednimi),</p> <p>Arkusze stylu CSS3 jako narzędzie budowania przyjaznego interfejsu użytkownika (m.in. zapytania typu Media Queries w CSS3),</p> <p>Wybrane narzędzia tworzenia rozbudowanych witryn internetowych na podstawie środowiska Google Sites ? charakterystyka i możliwości środowiska.</p> <p>Język XML w zastosowaniach internetowych ? składnia języka, definicja dokumentu poprawnego, definicja dokumentu zgodnego, SVG jako język budowania grafiki wektorowej w aplikacjach internetowych, KML jako język pozwalający na wykorzystywanie portali GIS do prezentowania danych przestrzennych, MathML jako język prezentacji rozbudowanych wzorów matematycznych.</p> <p>Idea Responsive Web Design ? budowanie witryn uniwersalnych, przeznaczonych na urządzenia mobilne i tradycyjne komputery stacjonarne.</p> <p>Języki skryptowe (na przykładzie JavaScript) tworzenia aplikacji typu client site processing. Wprowadzenie do techniki AJAX ? asynchronicznej metody budowania zawartości witryny internetowej.</p> <p>ASP.NET WebForms i ASP.NET WebPages jako przykłady technologii tworzenia aplikacji internetowych typu server site(kontrolki podstawowe, Master Page, kontrolki do nawigowania po witrynie, kontrolki walidacji, przekazywanie parametrów między stronami, kontrolki dostępu do baz danych, kontrolki autoryzacji dostępu do chronionych części witryny).</p> <p>Zajęcia laboratoryjne prowadzone są w formie dziesięciu 2-godzinnych ćwiczeń, odbywających się w laboratorium komputerowym wyposażonym w komputery ze stałym dostępem do Internetu. Zadania cząstkowe realizowane są przez studentów indywidualnie. Program laboratorium obejmuje następujące zagadnienia:</p> <p>Projekt rozbudowanej witryny internetowej w oparciu o środowisko Google Sites, w której zbierane będą następnie rozwiązania pozostałych zadań cząstkowych. Wykorzystywanie mechanizmów badania statystyki odwiedzin strony. Tworzenie przykładowej statycznej strony internetowej wykorzystującej nowe elementy strukturalne i multimedialne języka HTML 5. Przykład realizacji idei Responsive Web Design za pomocą elementu &lt;div&gt; języka HTML i zapytań Media Queries w CSS3. Budowanie złożonej grafiki wektorowej w języku SVG. Elementy animacji w języku SVG. Język matematyki na stronach WWW ? przykładowe rozwiązania. Budowanie wirtualnych wycieczek w języku KML i przy użyciu wtyczki programu Google Earth. Pisanie skryptów w języku JavaScript z wykorzystaniem elementów DOM ? projekt prostej gry. Animacja i grafika rastrowa z wykorzystaniem elementu &lt;canvas&gt; i języka JavaScript. Projekt przykładowych składowych aplikacji internetowej wykorzystującej kontrolki WebForms Asp.Net. Budowa prostej aplikacji AJAX z wykorzystaniem JavaScript i ASP.NET WebForms. Projekt przykładowej aplikacji internetowej wykorzystującej bazę danych w technologii ASP.NET WebPages.</p> <p>Metody dydaktyczne:</p> <ol style="list-style-type: none"> <li>1. wykład: prezentacja multimedialna, demonstracja wybranych technologii</li> <li>2. ćwiczenia laboratoryjne: rozwiązywanie zadań cząstkowych (niewielkie statyczne strony WWW i aplikacje internetowe), tworzenie witryny obejmującej rozwiązania wszystkich zadań cząstkowych</li> </ol>		
<p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"> <li>1. HTML5. Programowanie aplikacji, Zachary Kessin, Helion 2012</li> <li>2. CSS3 Tworzenie nowoczesnych stron WWW, Łukasz Pasternak, Helion 2012</li> <li>3. JavaScript. Aplikacje WWW, Alex MacCaw, Helion 2012</li> <li>4. ASP.NET WebForms, praca zbiorowa, Helion 2014</li> </ol>		
<p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"> <li>1. Responsive Web Design. Projektowanie elastycznych witryn w HTML5 i CSS3, Ben Frain, Helion 2013</li> </ol>		
<p><b>Bilans nakładu pracy przeciętnego studenta</b></p>		
<p><b>Czynność</b></p>		<p><b>Czas (godz.)</b></p>
1. udział w zajęciach laboratoryjnych		20
2. przygotowanie do ćwiczeń laboratoryjnych		10
3. udział w konsultacjach (mogą być realizowane drogą elektroniczną)związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu		4
4. dokończenie projektu witryny internetowej, uruchomienie i testy (czas poza zajęciami laboratoryjnymi)		12
5. udział w wykładach		10
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron		12
7. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym (10+2)		
<p><b>Obciążenie pracą studenta</b></p>		
<p><b>forma aktywności</b></p>	<p><b>godzin</b></p>	<p><b>ECTS</b></p>

Łączny nakład pracy	84	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	36	1
Zajęcia o charakterze praktycznym	42	2