



Akademia Nauk Stosowanych
im. Hipolita Cegielskiego w Gnieźnie Uczelnia Państwowa

SYLABUS

Pozycja przedmiotu w planie:		R.III/S.V - 6
1. OGÓLNY OPIS PRZEDMIOTU		
1	Nazwa modułu	Moduł zajęć kierunkowych
2	Nazwa przedmiotu	Inżynieria oprogramowania II
3	Kierunek studiów	Informatyka
4	Poziom studiów	studia pierwszego stopnia
5	Forma studiów	stacjonarne
6	Profil studiów	praktyczny
7	Rok studiów	trzeci
8	Semestr przedmiotu	piąty
9	Jednostka prowadząca kierunek studiów	Instytut Nauk Technicznych
10	Liczba punktów ECTS	3
11	Sposób zaliczenia:	Wykład: zaliczenie z oceną Projekt: zaliczenie z oceną
12	Imię i nazwisko nauczyciela (li) akademickiego (ich), stopień lub tytuł naukowy, adres e-mail	dr inż. Piotr Sujka p.sujka@ans-gniezno.edu.pl
13	Imię i nazwisko koordynatora(ów) przedmiotu, stopień lub tytuł naukowy, adres e-mail	dr inż. Piotr Sujka p.sujka@ans-gniezno.edu.pl
14	Język wykładowy	polski
15	Tryb prowadzenia zajęć	W salach i zdalny
16	Sposób prowadzenia zajęć	Synchroniczny i asynchroniczny
17	Narzędzia informatyczne wykorzystywane do prowadzenia zajęć, udostępniania materiałów i komunikacji ze studentami	Platforma Moodle. MS Teams
18	Przedmioty wprowadzające	Inżynieria oprogramowania I
19	Wymagania wstępne	1. Podstawowe wiadomości z zakresu technik programowania 2. Umiejętność efektywnego samokształcenia w dziedzinach związanych z informatyką jako wybranym kierunkiem studiów 3. Student ma świadomość konieczności poszerzania swoich kompetencji oraz gotowość do podjęcia współpracy w ramach zespołu.
20	Cele przedmiotu:	
C1	Poznanie zasad i metod wytwarzania oprogramowania w sposób systematyczny. Poznanie zasad zarządzania zespołami programistycznymi.	
C2	Student przyswaja sobie umiejętności: posługiwania się wzorcami projektowymi; projektowania oprogramowania zgodnie z metodyką obiektową.	

C3	Dokonywanie przeglądu projektu oprogramowania.	
21	Forma zajęć, liczba godzin wymagająca bezpośredniego udziału nauczyciela akademickiego, liczba godzin nakładu pracy studenta	
	Forma zajęć	Liczba godzin
	1. Wykład	30
	2. Projekt	15
	Suma godzin	45
lp.	Całkowity nakład pracy studenta	
	Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi:	Godzinowe obciążenie studenta
1.	Udział w wykładach 30 godzin Udział w projektach 15 godzin	45 godzin
	Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi 60 godzin, co odpowiada 1,5 punktom ECTS.	
2	Bilans nakładu pracy studenta: Samodzielne studiowanie literatury: 15 godzin Przygotowanie do zaliczenia z: 5 godzin Wykonywanie projektów: 15 godzin Łączny nakład pracy studenta wynosi 35 godzin, co odpowiada 1,5 punktom ECTS.	35 godzin
3	Łączny nakład pracy studenta (pozycja 1+2)	80 godzin
4	Punkty ECTS za przedmiot	3 ECTS
5	Liczba punktów ECTS, którą student musi osiągnąć w ramach zajęć o charakterze praktycznym w tym zajęć laboratoryjnych, warsztatowych, projektowych	1 ECTS
Efekty uczenia się - wiedza	<p>K_W17: Student ma wiedzę w zakresie projektowania oprogramowania, narzędzi i środowisk wytwarzania oprogramowania, procesów wytwarzania oprogramowania; ma wiedzę w zakresie specyfikacji wymagań, walidacji i testowania oprogramowania, zna metody zarządzania przedsięwzięciami programistycznymi oraz ich jakością.</p> <p>K_W18: Student ma wiedzę o cyklu życia oprogramowania, oraz cyklu życia układów cyfrowych, sprzętu komputerowego, sieciowego i systemów wbudowanych, ma wiedzę o stanie oraz najnowszych trendach i uwarunkowaniach rozwojowych informatyki, elektroniki, automatyki, robotyki i systemów ICT (IoT, BIoT, IIoT).</p> <p>K_W19: Student ma podstawową wiedzę w zakresie budowy relacyjnych i nierelacyjnych systemów baz danych, modelowania danych, projektowania relacyjnych i nierelacyjnych bazy danych, języków zapytań do baz danych, definicji danych oraz przetwarzania transakcji.</p>	
Efekty uczenia się - umiejętności	<p>K_U07: Student potrafi utworzyć specyfikację, zaprojektować i zaimplementować system informatyczny, cyfrowy lub wbudowany z zastosowaniem wybranych narzędzi wspierających budowę oprogramowania, wzorców projektowych, zgodnie z opracowanym harmonogramem.</p> <p>K_U08: Student ma umiejętności: posługiwania się wzorcami projektowymi; projektowania oprogramowania zgodnie z metodyką obiektową; dokonywania przeglądu projektu oprogramowania; wybierania narzędzi wspomagających budowę oprogramowania; doboru modelu procesu wytwarzania oprogramowania do specyfiki przedsięwzięcia; specyfikowania wymagań dotyczących oprogramowania i przeprowadzania ich przeglądu; tworzenia, oceny i realizacji planu testowania; uczestniczenia w inspekcji kodu; zarządzania konfiguracją oprogramowania; opracowywania planu przedsięwzięcia dotyczącego budowy oprogramowania.</p>	
Efekty uczenia się – kompetencje społeczne	K_K01: Student rozumie potrzebę i zna możliwości ciągłego dokształcania się, krytycznie odnosi się do posiadanej wiedzy, podnoszenia kompetencji zawodowych, osobistych i	

	<p>społecznych.</p> <p>K_K02: Student ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera-informatyka, w tym jej wpływ na środowisko i związaną z tym odpowiedzialność za podejmowane decyzje; dba o dobre tradycje zawodu informatyka.</p> <p>K_K04: Student ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania, potrafi określić priorytety służące realizacji określonego przez siebie lub zespół zadania.</p>
--	--

2. TREŚCI PROGRAMOWE ODNIESIONE DO EFEKTÓW UCZENIA SIĘ		
Forma zajęć	Treści programowe	liczba godzin
Forma: Wykład		
W1	Zarządzanie przedsięwzięciem programistycznym.	5
W2	Szacowanie złożoności, czasu i rozmiaru produktu.	5
W3	Metodyki tworzenia oprogramowania, ewolucja oprogramowania.	5
W4	Projektowanie obiektowe.	5
W5	Przegląd wybranych technologii programowych.	5
W6	Zarządzanie zespołami programistów.	5
Forma: Projekt		
P1	Identyfikacja klas i obiektów.	3
P2	Karty CRC.	3
P3	Przypadki użycia.	3
P4	Projekt struktur i hierarchii klas.	3
P5	Testowanie oprogramowania obiektowego.	3

3. Literatura	
Literatura podstawowa	<ol style="list-style-type: none"> Sommerville I., <i>Inżynieria oprogramowania</i>. WNT, Warszawa 2005. Sacha, <i>Inżynieria oprogramowania</i>, PWN, 2010.
Literatura uzupełniająca	<ol style="list-style-type: none"> Pressman R., <i>Praktyczne podejście do inżynierii oprogramowania</i>. WNT, Warszawa 2004. Jaskiewicz A., <i>Inżynieria oprogramowania</i>. Helion, Gliwice 1997

4. Metody dydaktyczne	
Forma	Metody dydaktyczne
Wykład	Wykład informacyjny (prezentacje multimedialne, materiały na platformie Moodle).
Projekt	Projekt, dyskusja.

5. Metody i kryteria oceniania															
Forma zajęć:	Forma zaliczenia:														
<p>Uzyskane punkty są przeliczane na oceny według następującej skali:</p> <table border="0"> <tr> <td>Procent punktów</td> <td>Ocena</td> </tr> <tr> <td>91-100%</td> <td>Bardzo dobry</td> </tr> <tr> <td>85-90%</td> <td>Dobry plus</td> </tr> <tr> <td>76-84%</td> <td>Dobry</td> </tr> <tr> <td>66-75%</td> <td>Dostateczny plus</td> </tr> <tr> <td>51-65%</td> <td>Dostateczny</td> </tr> <tr> <td>0-50%</td> <td>Niedostateczny</td> </tr> </table>		Procent punktów	Ocena	91-100%	Bardzo dobry	85-90%	Dobry plus	76-84%	Dobry	66-75%	Dostateczny plus	51-65%	Dostateczny	0-50%	Niedostateczny
Procent punktów	Ocena														
91-100%	Bardzo dobry														
85-90%	Dobry plus														
76-84%	Dobry														
66-75%	Dostateczny plus														
51-65%	Dostateczny														
0-50%	Niedostateczny														
<p>Opis:</p> <p>Forma zaliczenia: wykłady Ocena na podstawie prac pisemnych (kolokwia zaliczeniowe).</p> <p>Forma zaliczenia: projekty Zaliczenie końcowe na podstawie wykonania poszczególnych etapów projektu i zadań. Wykonanie poszczególnych zadań jest oceniane na bieżąco po zakończeniu ćwiczenia.</p>															
<p>Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen z każdej z form zajęć</p>															

	Zatwierdzenie karty opisu zajęć	
	<p>Stanowisko Tytuł/stopień naukowy, imię nazwisko</p>	Podpis
Opracował	dr inż. Piotr Sujka	
Zatwierdził	Dyrektor Instytutu Nauk Technicznych dr inż. Łukasz Józefowski	