

KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Inżynieria oprogramowania, C12
Nazwa przedmiotu (j. ang.):	Software engineering
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	Sieciowe Systemy Informatyczne, Technologie internetowe i bazy danych, Informatyka praktyczna, Bezpieczeństwo systemów informatycznych
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne/ studia niestacjonarne
Obszar kształcenia:	nauki techniczne (wg wykazu)
Dziedzina:	nauki techniczne (wg wykazu)
Dyscyplina nauki:	informatyka
Koordinator przedmiotu:	Dr inż. Agnieszka Kubacka

2. Ogólna charakterystyka przedmiotu

Przynależność do modułu:	kształcenia kierunkowego
Status przedmiotu:	obowiązkowy
Język wykładowy:	polski
Rok studiów, semestr:	II, 4
Forma i wymiar zajęć według planu studiów:	stacjonarne - wykład 30 h, ćw. Projektowe 15 h
Interesariusze i instytucje partnerskie (nieobowiązkowe)	
Wymagania wstępne:	Programowanie I, Programowanie II, Bazy danych, Algorytmy i struktury danych

3. Bilans punktów ECTS

Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt =25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami):	4 (A + B)	stacjonarne
A. Liczba godzin wymagających bezpośredniego udziału nauczyciela z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiągniętych na tych zajęciach	Obecność na wykładach Obecność na ćwiczeniach projektowych Egzamin W sumie: ECTS	30 15 2 47 1,5
B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS	przygotowanie projektu praca na platformie e-learningowej uzupełnienie/studiowanie notatek przygotowanie do egzaminu studiowanie zalecanej literatury w sumie: ECTS	35 10 5 15 10 75 2,5
C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS	udział w ćwiczeniach projektowych praca praktyczna samodzielna w sumie: ECTS	15 45 60 3

4. Opis przedmiotu

<p>Cel przedmiotu: Celem przedmiotu jest zapoznanie studentów z metodyką tworzenia oprogramowania. Zostanie im przedstawiony cykl życia oprogramowania oraz opis poszczególnych faz tego cyklu na przykładzie wybranych modeli. Omówione zostaną narzędzia CASE.</p>
<p>Metody dydaktyczne: wykład, ćwiczenia projektowe</p>
<p>Treści kształcenia:</p> <p>Wykłady: Wprowadzenie do inżynierii oprogramowania. Modele cyklu życia oprogramowania. Fazy modelu kaskadowego. Metody, techniki i narzędzia inżynierii wstecznej.</p> <p>Ćwiczenia projektowe: Studenci pracują w zespołach 2 – 4 osobowych nad wybranymi projektami. Ich tematyka uzgadniana jest z prowadzącym zajęcia. Podczas kolejnych zajęć realizują i prezentują cząstkowe wyniki swojej pracy, będące kolejnymi etapami przedsięwzięcia programistycznego. Jako końcowy efekt pracy zespołu jest gotowy projekt wraz z dokumentacją przygotowaną zgodnie z wytycznymi obowiązującymi w inżynierii oprogramowania.</p>

5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia				
K_W06, K_W07, K_W08, K_W16, K_U04, K_U17, K_U18, K_U22, K_U28, K_K01, K_K03, K_K04, K_K05				
Efekt przedmiotu (kod przedmiotu + kod efektu kształcenia)	Student, który zaliczył przedmiot (spełnił minimum wymagań)			Efekt kierunkowy
C12_K_W07 C12_K_W16	Wiedza: 1. Ma podstawową wiedzę o cyklu życia i trendach rozwojowych systemów informatycznych sprzętowych lub programowych. 2. Zna wzorce projektowe stosowane w projektowaniu aplikacji. Zna metody wytwarzania oprogramowania i techniki stosowane w ramach metod.			K_W07 K_W16
C12_K_U04 C12_K_U18 C12_K_U22	Umiejętności: 1. Potrafi pracować indywidualnie i w zespole informatyków, w tym także potrafi zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów. 2. Potrafi zaprojektować poprawny interfejs użytkownika dla aplikacji, w tym internetowych. 3. Potrafi utworzyć specyfikację, zaprojektować i zaimplementować system informatyczny z zastosowaniem wybranych narzędzi wspierających budowę oprogramowania, wzorców projektowych i zgodnie z opracowanym harmonogramem.			K_U04 K_U18 K_U22
C12_K_K01 C12_K_K04 C12_K_K05	Kompetencje społeczne: 1. Rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe. 2. Jest odpowiedzialny za pracę własną i skutki podejmowanych decyzji, potrafi podporządkować się zasadom pracy w grupie w roli lidera i członka zespołu, jest odpowiedzialny za wspólnie realizowane zadania. 3. Potrafi wykazać się skutecznością w realizacji projektów o charakterze społecznym, naukowo-badawczym lub programistyczno-wdrożeniowym, wchodzących w program studiów lub realizowanych poza studiami.			K_K01 K_K04 K_K05
Sposoby weryfikacji efektów kształcenia:				
Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca – przykładowe sposoby jej wystawienia poniżej	Ocena końcowa – przykładowe sposoby jej wystawienia

				poniżej
1	C12_K_W06 C12_K_W07 C12_K_W08 C12_K_W16	Aktywność podczas zajęć, poprawność i terminowość wykonywania poszczególnych etapów projektu, prezentacja projektu. Egzamin.	Ocena z prezentacji i projektu, ocena z dokumentacji do projektu,	Średnia z ocen formujących, ocena z egzaminu
2	C12_K_U04 C12_K_U18 C12_K_U22	Aktywność podczas zajęć, poprawność i terminowość wykonywania poszczególnych etapów projektu, prezentacja projektu. Egzamin.	Ocena z projektu, ocena z dokumentacji do projektu	Średnia z ocen formujących, ocena z egzaminu
3	C12_K_K01 C12_K_K04 C12_K_K05	Aktywność podczas zajęć, poprawność i terminowość wykonywania poszczególnych etapów projektu, prezentacja projektu. Egzamin.	Ocena z projektu, ocena z dokumentacji do projektu	Średnia z ocen formujących, ocena z egzaminu

Kryteria oceny (oceny 3,0 powinny być równoważne z efektami kształcenia, choć mogą być bardziej szczegółowo opisane):

w zakresie wiedzy		Efekt kształcenia
Na ocenę 3,0	Przygotował projekt wraz z dokumentacją opierający się na jednym z modeli cyklu życia oprogramowania, który potrafi omówić. Prawidłowo dobrał narzędzie do wytworzenia programu.	C12_K_W07 C12_K_W16
Na ocenę 5,0	Przygotował projekt z dokumentacją opierający się na jednym z modeli cyklu życia oprogramowania. Potrafi omówić inne modele cyklu życia oprogramowania, niż użyty w projekcie. Prawidłowo dobrał narzędzie do wytworzenia programu. Potrafi zaproponować inne narzędzia, jakie można wykorzystać przy tworzeniu projektu.	C12_K_W07 C12_K_W16
w zakresie umiejętności		
Na ocenę 3,0	Projekt został wykonany i oddany w ustalonym terminie. Student potrafi wymienić podstawowe problemy, które napotkał podczas tworzenia projektu, wskazuje i objaśnia części projektu, za które był odpowiedzialny. Omawia cechy interfejsu użytkownika.	C12_K_U04 C12_K_U18 C12_K_U22
Na ocenę 5,0	Projekt został wykonany zgodnie z opracowanym harmonogramem oraz oddany w ustalonym terminie. Student potrafi wymienić problemy, które zostały napotkane przez członków zespołu podczas tworzenia projektu, wskazuje i objaśnia całość projektu. Omawia cechy interfejsu użytkownika, w tym wskazuje cechy, które czynią go odpornym na błędy popełniane przez użytkownika. Potrafi wskazać inne narzędzia, jakie mogły zostać użyte do przygotowania projektu.	C12_K_U04 C12_K_U18 C12_K_U22
w zakresie kompetencji społecznych		
Na ocenę 3,0	Wie, jak zmieniały się narzędzia programistyczne. Potrafi wskazać inne narzędzie, przy pomocy którego można rozwiązać zadany problem. Potrafi współpracować z pozostałymi członkami zespołu.	C12_K_K01 C12_K_K04 C12_K_K05
Na ocenę 5,0	Wie, jak zmieniały się narzędzia programistyczne. Potrafi wskazać inne narzędzia, przy pomocy których można rozwiązać zadany problem. Potrafi współpracować z pozostałymi członkami zespołu, a w razie potrzeby przejąć rolę lidera grupy. Potrafi identyfikować aspekty prawne i społeczne tworzenia oprogramowania.	C12_K_K01 C12_K_K04 C12_K_K05

Kryteria oceny końcowej:

Egzamin:

ocena z egzaminu: 100%

Ćwiczenia projektowe:

Prezentacja(omówienie) projektu: 10%

wykonanie projektu: 30%

ocena wykonanej dokumentacji: 60%

Zalecana literatura:

1. Jaskiewicz A., Inżynieria oprogramowania, Helion , Gliwice, 1997
2. Bass L. Architektura oprogramowania w praktyce, Helion, Gliwice, 2011
3. Zmitrowicz, K., Jakość projektów informatycznych, Helion, Gliwice, 2015
4. Kan S.H., Metryki i modele w inżynierii jakości oprogramowania, Wydawnictwo Naukowe PWN, Warszawa, 2006

Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:
Przygotowanie do wykładów i ćwiczeń projektowych – 30 godzin
Konsultacje – 15 godzin
Poprawa dokumentacji projektów– 40 godzin
Przygotowanie i poprawa egzaminu – 5 godzin
W sumie: 90 godzin