

KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Tworzenie bezpiecznego kodu D1_13
Nazwa przedmiotu (j. ang.):	Creating secure code
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	Bezpieczeństwo systemów informatycznych
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne
Obszar kształcenia:	nauki techniczne
Dziedzina:	nauki techniczne
Dyscyplina nauki:	informatyka
Koordinator przedmiotu:	Dr inż. Bartosz Trybus

2. Ogólna charakterystyka przedmiotu

Przynależność do modułu:	kształcenia specjalnościowego
Status przedmiotu:	obowiązkowy
Język wykładowy:	Polski
Rok studiów, semestr:	III, 6
Forma i wymiar zajęć według planu studiów:	stacjonarne - wykład 15 h, ćw. laboratoryjne 15 h
Interesariusze i instytucje partnerskie (nieobowiązkowe)	
Wymagania wstępne / Przedmioty wprowadzające:	Programowanie niskopoziomowe, Programowanie I, II, Bazy danych, Języki i paradygmaty programowania

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):	3	stacjonarne
A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontaktowych, w czasie rzeczywistym, w tym testy, egzaminy etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiąganych na tych zajęciach	obecność na wykładach obecność na ćwiczeniach laboratoryjnych ćwiczenia projektowe udział w konsultacjach W sumie: ECTS	15 15 5 35 1,2
B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS (np. praca w bibliotece, w sieci, na platformie e-learningowej, w laboratorium, praca nad projektem końcowym, przygotowanie ogólne; suma poszczególnych godzin powinna zgadzać się z liczbą ogólną)	przygotowanie ogólne opracowanie dokumentacji (sprawozdań) praca nad projektem studiowanie zalecanej literatury praca w sieci w sumie: ECTS	5 10 10 10 35 1,8
C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS (ta liczba nie musi być powiązana z liczbą godzin kontaktowych, niektóre zajęcia praktyczne/laboratoryjne mogą odbywać się bez udziału nauczyciela):	udział w zajęciach praca samodzielna w sumie: ECTS	15 22 37 1,5

4. Opis przedmiotu

<p>Cel przedmiotu: Zdobycie wiedzy i umiejętności praktycznych z zakresu programowania bezpiecznych systemów informatycznych.</p>
<p>Metody dydaktyczne: wykład informacyjny, ćwiczenia laboratoryjne, symulacja</p>
<p>Treści kształcenia: Wykłady i laboratoria: Zasady i techniki zabezpieczeń. Uwierzytelnianie użytkownika. Ochrona tajnych danych. Projektowanie systemów zabezpieczeń. Techniki pisania kodu. Zabezpieczenia przed przepełnieniem bufora. Korzystanie z funkcji kryptograficznych. Walidacja danych w aplikacjach webowych. Zabezpieczenie przez atakami typu wstrzykiwanie kodu.</p>

Poziomy uprzywilejowania kodu. Implementacja kontroli dostępu.
 Zabezpieczenie komunikacji sieciowej z architekturą klient-serwer. Ochrona przed atakami typu DDoS.
 Wybrane zagadnienia zabezpieczeń. Obsługa błędów i wyjątków. Asercje. Testowanie kodu. Przegląd technik weryfikacyjnych.

5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia (w sumie wymienić ok. od 3 do 9 efektów - podać numery efektów z listy dla danego kierunku/specjalności – opublikowane na stronie uczelni; podać TYLKO te efekty (*tam gdzie to możliwe i stosowne w trzech kategoriach*, np. kompetencje społeczne mogą nie być realizowane w tym przedmiocie), na których osiągnięcie kładzie się nacisk w ramach przedmiotu, wybrane efekty kierunkowe powinny być bardziej szczegółowo sformułowane niż te dla całej specjalności, tak aby były weryfikowalne – dlatego mają osobne symbole jako efekty przedmiotu)

Efekt przedmiotu (kod przedmiotu + kod efektu kształcenia)	Student, który zaliczył przedmiot (spełnił minimum wymagań)	Efekt kierunkowy
D1_13_K_W01 D1_13_K_W02 D1_13_K_W03	Wiedza: 1. Zna podstawowe zasady tworzenia zabezpieczeń w programach, w szczególności aplikacjach webowych. 2. Zna techniki zabezpieczeń programu przed typowymi zagrożeniami, w tym przed wstrzykiwaniem kodu. 3. Zna zagadnienia związane z uprzywilejowaniem kodu, ochroną komunikacji między procesami i obsługą błędów.	K_W05 K_W06 K_W08 K_W12
D1_13_K_U01 D1_13_K_U02 D1_13_K_U03	Umiejętności 1. Potrafi zaimplementować mechanizm autoryzacji w aplikacji, w tym webowej. 2. Potrafi użyć w programie mechanizmy kryptograficzne do obsługi wrażliwych danych. 3. Potrafi zaimplementować wielopoziomą walidację danych i obsługiwać sytuacje nadzwyczajne: błędy, wyjątki.	K_U02 K_U11 K_U15 K_U25
D1_13_K_K01	Kompetencje społeczne 1. Potrafi pracować w zespole projektującym i implementującym system zabezpieczeń w oprogramowaniu.	K_K04 K_K08

Sposoby weryfikacji efektów kształcenia:

(np. dyskusja, gra dydaktyczna, zadanie e-learningowe, ćwiczenie laboratoryjne, projekt indywidualny/ grupowy, zajęcia terenowe, referat studenta, praca pisemna, kolokwium, test zaliczeniowy, egzamin, opinia eksperta zewnętrznego, etc. Dodać do każdego wybranego sposobu symbol zakładanego efektu, jeśli jest ich więcej)

Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca – przykładowe sposoby jej wystawienia poniżej	Ocena końcowa przykładowe sposoby jej wystawienia poniżej
1	D1_13_K_W01 D1_13_K_W02 D1_13_K_W03 D1_13_K_U01 D1_13_K_U02 D1_13_K_U03	Zaliczenie	sprawdzian wiedzy, sprawdzian umiejętności	rozwiązanie zadania problemowego (zadań), analiza przypadku
2	D1_13_K_U01 D1_13_K_U02	ćwiczenia laboratoryjne	kolokwia zaliczające partie mate-	średnia z ocen formujących

	D1_13_K_U03 D1_13_K_K01		riału, rozwiązania zadań domowych		
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	Student uzyskał min. 50% wymaganej wiedzy w zakresie obowiązującego materiału. Student:	1. Zna obszary potencjalnie zagrożone pod względem bezpieczeństwa przy tworzeniu oprogramowania,	2. Zna wybraną technikę zabezpieczenia przed wstrzykiwaniem kodu,	3. Zna zasady uruchamiania kodu na najniższym poziomie uprzywilejowania.	D1_13_K_W01 D1_13_K_W02 D1_13_K_W03
Na ocenę 5,0	Student zdobył powyżej 95% wymaganej wiedzy w zakresie obowiązującego materiału. Student:	Zna techniki zabezpieczania oprogramowania przez różnymi zagrożeniami.			D1_13_K_W01 D1_13_K_W02 D1_13_K_W03
w zakresie umiejętności					
Na ocenę 3,0	Student uzyskał min. 50% wymaganych umiejętności w zakresie obowiązującego materiału. Student potrafi:	1. Zaprogramować mechanizm autoryzacji w aplikacji webowej	2. Użyć funkcji skrótu.	3. Obsługiwać wyjątki w języku programowania.	D1_13_K_U01 D1_13_K_U02 D1_13_K_U03
Na ocenę 5,0	Student uzyskał powyżej 95% umiejętności w zakresie obowiązującego materiału. Student umie:	Zaprojektować i zaimplementować oprogramowanie z kompleksowym uwzględnieniem zasad bezpieczeństwa.			D1_13_K_U01 D1_13_K_U02 D1_13_K_U03
w zakresie kompetencji społecznych					
Na ocenę 3,0	Student osiągną wymagane kompetencje społeczne na poziomie min. 50%. Potrafi pracować w zespole projektującym system informatyczny składający się z warstw danych, logiki i prezentacji.				D1_13_K_K01
Na ocenę 5,0	Student osiągną wymagane kompetencje społeczne na poziomie wyższym niż 90%.	1. Potrafi pracować w zespole projektującym i implementującym system zabezpieczeń w oprogramowaniu.	2. Jest w stanie pełnić rolę kierownika zespołu realizującego zadanie projektowe.		D1_13_K_K01
Zalecana literatura (w podziale na literaturę podstawową i uzupełniającą):					
Podstawowa:					
1. Strona WWW: emateriały.pwsz.krosno.pl.					
2. C# 6.0 i MVC 5 : tworzenie nowoczesnych portali internetowych / Krzysztof Żydzik, Tomasz Rak					
3. TDD [Test-Driven Development] : programowanie w Javie sterowane testami : naucz się podstaw metodyki TDD / Viktor Farcic, Alex Garcia					
4. Visual Studio 2013 : podręcznik programowania w C# z zadaniami / Jacek Matulewski					
Uzupełniająca:					
1. Java : techniki zaawansowane / Cay S. Horstmann, Gary Cornell					

Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin: (np. indywidualne konsultacje, poprawa prac, przygotowanie projektu zaliczeniowego, egzaminu, przygotowanie ćwiczeń e-learningowych). Przykład poniżej

Aktualizacja wykładów, ćwiczeń i zadań domowych – 30 godzin

Ocena sprawozdań i zadań domowych – 20 godzin

Konsultacje – 10 godzin

Przygotowanie zadań zaliczeniowych – 5 godzin

W sumie: 65 godzin