

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu zajęć:	Inżynieria oprogramowania				
Rok akademicki:	2019/2020	Kod:	HNKT-1-509-s	Punkty ECTS:	3
Wydział:	Humanistyczny				
Kierunek:	Nowoczesne technologie w kryminalistyce	Specjalność:	—		
Poziom studiów:	Studia I stopnia	Forma studiów:	Stacjonarne		
Język wykładowy:	Polski	Profil:	Ogólnoakademicki (A)	Semestr:	5
Strona www:	—				
Prowadzący moduł:	dr inż. Kołodziej Jacek (jackolo@agh.edu.pl)				

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Poznanie metodyk projektowania i wytwarzania oprogramowania.

Opis efektów uczenia się dla modułu zajęć

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Powiązania z KEU	Sposób weryfikacji i oceny efektów uczenia się osiągniętych przez studenta w ramach poszczególnych form zajęć i dla całego modułu zajęć
Wiedza: zna i rozumie			
M_W001	Student ma wiedzę w zakresie budowy systemów informatycznych, metodyk oraz stosowanych narzędzi	NKT1A_W05	Projekt, Kolokwium, Aktywność na zajęciach
Umiejętności: potrafi			
M_U001	Student potrafi opracować dokumentację dotyczącą realizacji prostego procesu budowy oprogramowania	NKT1A_U04, NKT1A_U07, NKT1A_U06	Projekt, Kolokwium, Aktywność na zajęciach
M_U002	Student ma umiejętności pozwalające na zaprojektowanie procesu budowy oprogramowania zgodnie ze specyfikacją wymagań	NKT1A_U04, NKT1A_U07, NKT1A_U06	Projekt, Kolokwium, Aktywność na zajęciach
Kompetencje społeczne: jest gotów do			
M_K001	Student potrafi pracować w grupie projektowej, zajmując w niej różne role	NKT1A_K02	Wykonanie projektu, Aktywność na zajęciach

Liczba godzin zajęć w ramach poszczególnych form zajęć

Suma	Forma zajęć dydaktycznych										
	Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
48	20	0	0	28	0	0	0	0	0	0	0

Matryca kierunkowych efektów uczenia się w odniesieniu do form zajęć i sposobu zaliczenia, które pozwalają na ich uzyskanie

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Forma zajęć dydaktycznych										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
Wiedza: zna i rozumie												
M_W001	Student ma wiedzę w zakresie budowy systemów informatycznych, metodyk oraz stosowanych narzędzi	+	-	-	-	-	-	-	-	-	-	-
Umiejętności: potrafi												
M_U001	Student potrafi opracować dokumentację dotyczącą realizacji prostego procesu budowy oprogramowania	-	-	-	+	-	-	-	-	-	-	-
M_U002	Student ma umiejętności pozwalające na zaprojektowanie procesu budowy oprogramowania zgodnie ze specyfikacją wymagań	-	-	-	+	-	-	-	-	-	-	-
Kompetencje społeczne: jest gotów do												
M_K001	Student potrafi pracować w grupie projektowej, zajmując w niej różne role	+	-	-	-	-	-	-	-	-	-	-

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w zajęciach dydaktycznych/praktyka	48 godz
Przygotowanie do zajęć	10 godz
przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	20 godz
Samodzielne studiowanie tematyki zajęć	5 godz
Egzamin lub kolokwium zaliczeniowe	2 godz
Dodatkowe godziny kontaktowe	5 godz
Sumaryczne obciążenie pracą studenta	90 godz
Punkty ECTS za moduł	3 ECTS

Pozostałe informacje

Szczegółowe treści kształcenia w ramach poszczególnych form zajęć (szczegółowy program wykładów i pozostałych zajęć)

Wykład

Projektowanie oprogramowania. Analiza ryzyka, zarządzanie ryzykiem i jakością: ekonomika jakości. Procesy wytwarzania oprogramowania. Wymagania i ich specyfikacja (standardy, wzorce), pozyskiwanie i weryfikacja wymagań, kontrola zmian. Walidacja i testowanie oprogramowania (testy automatyczne). Ewolucja oprogramowania. Zarządzanie przedsięwzięciem programistycznym. Projektowanie oprogramowania zgodne z metodyką strukturalną oraz obiektową, techniki obiektowego modelowania – standard UML. Narzędzia i środowiska wytwarzania oprogramowania (CASE, RAD, języki czwartej i piątej generacji). Zarządzanie konfiguracją oprogramowania. Zarządzanie wersjami. Standaryzacja i normalizacja procesu wytwarzania – norma ISO90003. Jakość oprogramowania: metryki oprogramowania, ocena wydajności oprogramowania. Wpływ otoczenia (sprzęt, oprogramowanie, ludzie) na systemy informatyczne, niezawodność oprogramowania, inspekcje oprogramowania. Model CMMI. Wielokrotne używanie kodu, korzystanie z API (Application Programming Interface).

Ćwiczenia projektowe

Laboratorium:

- L1. Metody analizy ryzyka, ocena ryzyka realizacji projektu
- L2. Planowanie, diagramy Ganta, zasobów, estymacja kosztów.
- L3. Pozyskiwanie i weryfikacja wymagań, pominięcia i niejednoznaczności
- L4. Formularze, notacje formalne, zastosowanie „ pseudokodu” PDL w inżynierii wymagań
- L5. Notacja UML w modelowaniu statycznych cech systemów informatycznych
- L6. Notacja UML w modelowaniu dynamicznych cech systemów informatycznych I
- L7. Modelowanie funkcjonalne, modele przepływu danych
- L8. Test – podsumowanie

Metody i techniki kształcenia:

Wykład: Treści prezentowane na wykładzie są przekazywane w formie prezentacji multimedialnej w połączeniu z klasycznym wykładem tablicowym wzbogaconymi o pokazy odnoszące się do prezentowanych zagadnień.

Ćwiczenia projektowe: Studenci wykonują zadany projekt samodzielnie, bez większej ingerencji prowadzącego. Ma to wykształcić poczucie odpowiedzialności za pracę w grupie oraz odpowiedzialności za podejmowane decyzje.

Inne metody: prezentacja multimedialna, gry symulacyjne, praca w grupach

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:

Nie określono

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa:

Wykład:

- Obecność obowiązkowa: Nie

- Zasady udziału w zajęciach: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia projektowe:

- Obecność obowiązkowa: Tak

- Zasady udziału w zajęciach: Studenci wykonują prace praktyczne mające na celu uzyskanie kompetencji zakładanych przez sylabus. Ocenie podlega sposób wykonania projektu oraz efekt końcowy.

Sposób obliczania oceny końcowej

Ocena końcowa jest oceną zajęć laboratoryjnych i wykładu

Składowe oceny:

1. udział w wykładach 15 × 1pkt lub kolokwium zaliczeniowe (max 15)
1. aktywny udział w zajęciach laboratoryjnych 8×5 pkt
2. wykonanie zadania zaliczeniowego 1 30pkt
3. wykonanie zadania zaliczeniowego 2 30pkt

Dopuszczalna jest jedna nieobecność.

Punkty przeliczane są na oceny zgodnie z regulaminem studiów.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:

Nie określono

Wymagania wstępne i dodatkowe, z uwzględnieniem sekwencyjności modułów

Nie podano wymagań wstępnych lub dodatkowych.

Zalecana literatura i pomoce naukowe

1. Sommerville Ian, Inżynieria oprogramowania, WNT
2. UML. Wprowadzenie
2. Bezier B., Software Testing Techniques

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

Nie podano dodatkowych publikacji

Informacje dodatkowe

Brak