

(pieczęć jednostki organizacyjnej)

KARTA PRZEDMIOTU

1) Nazwa przedmiotu: PROGRAMOWANIE OBIEKTOWE		2) Kod przedmiotu: S I-AiIP/17		
3) Karta przedmiotu ważna od roku akademickiego: 2017/2018				
4) Forma kształcenia: studia stacjonarne				
5) Poziom kształcenia: studia I stopnia				
6) Kierunek studiów: AUTOMATYKA I INFORMATYKA PRZEMYSŁOWA				
7) Profil studiów: praktyczny				
8) Specjalność:				
9) Semestr: 3				
10) Jednostka prowadząca przedmiot: Katedra Elektryfikacji i Automatykacji Górnictwa				
11) Prowadzący przedmiot: dr inż. Sergiusz Boron				
12) Przynależność do grupy przedmiotów: podstawowy				
13) Status przedmiotu: obowiązkowy				
14) Język prowadzenia zajęć: polski				
15) Przedmioty wprowadzające oraz wymagania wstępne: Przedmiotem niezbędnym , wprowadzającym, są Podstawy programowania . Przedmiotami uzupełniającymi są: Podstawy informatyki, Matematyka oraz Język angielski .				
16) Cel przedmiotu: Celem przedmiotu jest zapoznanie studentów z podstawami projektowania i programowania obiektowego. Studenci mają nabyć umiejętność w zakresie obiektowego podejścia do analizy, projektowania i realizacji zadań programistycznych. Celem przedmiotu nie jest nauka podstaw programowania, w związku z tym warunkiem niezbędnym do realizacji przedmiotu jest znajomość zagadnień teoretycznych i praktycznych z przedmiotu Podstawy programowania, który to przedmiot obejmuje niezbędne podstawy z zakresu programowania. Przedmiot stanowi kontynuację i rozszerzenie zagadnień związanych z programowaniem. Jako język programowania zostanie wykorzystany Język C++ (semestr III i IV) oraz Java (semestr IV).				
17) Efekty kształcenia:¹				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
1.	Ma wiedzę w zakresie technik obiektowej analizy i projektowania oprogramowania, rozumie różnicę między podejściem strukturalnym i obiektowym.	Kolokwium	Wykład	K_W04+++ K_U16+

¹ należy wskazać ok. 5 – 8 efektów kształcenia

2.	Posiada umiejętność w zakresie przeprowadzenia analizy i projektowania obiektowego dla wybranego zagadnienia o podstawowym i średnim stopniu złożoności.	Kolokwium, zadanie praktyczne	Wykład, laboratorium	K_W04+++ K_U08+++ K_U16+ K_U17+
3.	Posiada umiejętność w zakresie praktycznej realizacji programów, w wybranym środowisku i języku programowania, w oparciu o techniki obiektowe.	Zadanie praktyczne	Laboratorium	K_W04++ K_U01++ K_U08++ K_U17+
4.	Zaznajomiony jest, w zakresie podstawowym, z możliwościami i technikami kompleksowego tworzenia oprogramowania przy wykorzystaniu narzędzi programistycznych typu IDE/RAD/CASE (Integrated Development Environment/Rapid Application Development/Computer Assisted Software Engineering).	Zadanie praktyczne	Wykład, laboratorium	K_W04++ K_U08++ K_U12++
5.	Posiada podstawową wiedzę na temat architektury MVC (Model-View-Controller).	Zadanie praktyczne	Wykład, laboratorium	K_W04+ K_U08+
6.	Posiada podstawową wiedzę w zakresie inżynierii programowania.	Kolokwium	Wykład	K_W04+ K_U12+ K_U16+
7.	Posiada umiejętność korzystania z dokumentacji technicznych (również w języku angielskim).	Zadanie praktyczne	Laboratorium	K_W04++
8.	Posiada umiejętność w zakresie testowania, lokalizacji i eliminacji błędów tworzonego oprogramowania.	Zadanie praktyczne	Laboratorium, projekt	K_W04+ K_U08+ K_U12+

18) Formy zajęć dydaktycznych i ich wymiar (liczba godzin)

Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
10		15		

Treści kształcenia: (oddzielnie dla każdej z form zajęć dydaktycznych W./Ćw./L./P./Sem.)

Wykład

Wprowadzenie do technik analizy i projektowania obiektowego (Object Oriented Programming). Wprowadzenie do języka C++ (techniki obiektowe przedstawione są na przykładzie języka C++). Paradygmaty programowania obiektowego. Techniki strukturalne, a obiektowe. Klasy i obiekty. Dziedziczenie i hierarchia klas. Dziedziczenie wielobazowe. Klasy i funkcje wirtualne. Polimorfizm. Przeciążanie operatorów. Obsługa wyjątków. Wzorce (szablony). Dynamiczna i statyczna kontrola typów. Pliki i strumienie. Biblioteka STL, kontenery, łańcuchy znaków. Podstawy przetwarzania równoległego. Architektura MVC. Elementy inżynierii programowania. Narzędzia IDE/RAD/CASE. Tworzenie aplikacji w języku C++. Uruchamianie aplikacji, testowanie, lokalizacja i eliminacja błędów.

Laboratorium

Program zajęć laboratoryjnych jest ściśle powiązany z tematyką wykładów. Stanowi jego rozszerzenie i praktyczną realizację omawianych zagadnień w wybranym środowisku programistycznym. W ramach laboratorium realizowane będą programy i zadania ilustrujące i wyjaśniające zagadnienia omawiane w ramach wykładu.

19) Egzamin: NIE

20) Literatura podstawowa:

1. Weisfeld M.: *Myślenie obiektowe w programowaniu*. Wydawnictwo Helion.
2. Jaskiewicz A.: *Inżynieria oprogramowania*. Wydawnictwo Helion.
3. Porębski W.: *Programowanie w języku C++*. Komputerowa Oficyna Wydawnicza „HELP”.
4. Grębosz J.: *Symfonia C++ Standard. Programowanie w języku C++ orientowane obiektowo*. Tom I i II. Wydawnictwo Helion.
5. Stroustrup B.: *Programowanie. Teoria i praktyka z wykorzystaniem C++*. Wydawnictwo Helion.
6. Josuttis. N. M.: *C++. Projektowanie zorientowane obiektowo. Vademecum profesjonalisty*. Wydawnictwo Helion.
7. Borowik W., Borowik B.: *Meandry Języka C++*. Wydawnictwo Mikom.

21) Literatura uzupełniająca:

1. Strona internetowa: *MSDN (Microsoft Developer Network)*.
2. Strona internetowa: *Microsoft Visual Studio (C++)*.
3. Strona internetowa: *Embarcadero RAD Studio, C++ Builder*.
4. Wróblewski P.: *Algorytmy, struktury danych u techniki programowania*. Wydawnictwo Helion.
5. Stroustrup B.: *Język C++. Kompendium wiedzy*. Wydawnictwo Helion.
6. Dorobek M.: *C++ Builder. Podręcznik*. Wydawnictwo MIKOM.
7. Owczarek M.: *Microsoft Visual C++ 2008. Praktyczne przykłady*. Ebook. Wydawnictwo Helion.

22) Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1.	Wykłady	10/15 w tym: zapoznanie się ze wskazaną literaturą (4), przygotowanie do wykładów i kolokwium (10), kolokwium (1).
2.	Ćwiczenia	-
3.	Laboratorium	15/30 w tym: zapoznanie się ze wskazaną literaturą (1), uczestnictwo w konsultacjach (4), ćwiczenia własne w pisaniu kodu (24), praktyczne sprawdzenie wiedzy (1).
4.	Projekt	-
5.	Seminarium	-
6.	Inne	-
Suma godzin:		25/45
23. Suma wszystkich godzin:		70
24. Liczba punktów ECTS:		3
25. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego:		1
26. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty, ćwiczenia):		2
27. Uwagi:		
Przedmiot obejmuje dwa semestry 3 i 4. Karta dotyczy semestru 3. W semestrze 3 zajęcia prowadzone są w oparciu o język JAVA. Zajęcia projektowe w semestrze 4 realizowane są przy wykorzystaniu języka C++ lub JAVA – wybór studenta.		

Zatwierdzono:

.....
(data i podpis prowadzącego).....
(data i podpis Dyrektora/Kierownika podstawowej lub międzywydziałowej jednostki organizacyjnej)