

KARTA PRZEDMIOTU

Nazwa w języku polskim: **Informatyka II**

Nazwa w języku angielskim: **Software Engineering**

Kierunek studiów (jeśli dotyczy): **Robotyka i Automatyzacja Procesów**

Poziom i forma studiów: **I stopień, stacjonarne**

Rodzaj przedmiotu: **obowiązkowy**

Kod przedmiotu: **RAM031047**

Grupa kursów: **nie**

	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć zorganizowanych w Uczelni (ZZU)	15			30	
Liczba godzin całkowitego nakładu pracy studenta (CNPS)	30			60	
Forma zaliczenia	Zaliczenie na ocenę			Zaliczenie na ocenę	
Grupa kursów					
Liczba punktów ECTS	1			2	
w tym liczba punktów odpowiadająca zajęciom o charakterze praktycznym (P)				2	
w tym liczba punktów ECTS odpowiadająca zajęciom wymagającym bezpośredniego kontaktu (BK)	0.6			1.4	

WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I KOMPETENCJI SPOŁECZNYCH

1. Posiada podstawową wiedzę z zakresu algorytmów komputerowych
2. Zna semantykę i syntaktykę języka C

CELE PRZEDMIOTU

- C1. Wyjaśnić zadania, metody i narzędzia (UML) inżynierii oprogramowania, wprowadzić modelowanie obiektowe. Przygotować do praktycznych zajęć z programowania obiektowego
- C2. Nauczyć umiejętności stosowania proceduralnego paradygmatu programowania
- C3. Nauczyć umiejętności opracowywania programów realizujących zadania przetwarzania danych

PRZEDMIOTOWE EFEKTY UCZENIA SIĘ

I. Z zakresu wiedzy:

PEK_W01 - Rozumie istotę metodycznego rozwiązywania problemów programistycznych i stosowania narzędzi

PEK_W02 - Posiada wiedzę dotyczącą paradygmatu programowania obiektowego i zapisu UML

PEK_W03 - Potrafi czytać proste programy w języku C++

II. Z zakresu umiejętności:

PEK_U01 - Potrafi zastosować proceduralny paradygmat programowania, tj. rozbić problem programistyczny na zestaw funkcji realizujących poszczególne zadania w języku C

PEK_U02 - Potrafi zastosować język C do przetwarzania złożonych zbiorów danych jedno i dwu-wymiarowych z wykorzystaniem dynamicznych struktur danych

PEK_U03 - Potrafi testować i debugować programy pisane w języku C, korzystać z dokumentacji bibliotek języka C, zna i stosuje zasady poprawnego stylu programowania

III. Z zakresu kompetencji społecznych:

PEK_K01 - Potrafi wyszukiwać oraz krytycznie analizować informacje

PEK_K02 - Potrafi zaplanować zadanie programistyczne, wykonać zgodnie z wymaganiami funkcjonalnymi i odbiorów

TREŚCI PROGRAMOWE

Forma zajęć – Wykład		Liczba godzin
Wy1	Wprowadzenie do obiektowości, przykłady, narzędzia CASE	2
Wy2	Modele rozwoju oprogramowania (cykle życia)	2
Wy3	Obiektowa analiza wymagań (UML - diagramy przypadków użycia, aktywności)	2
Wy4	Obiektowe projektowanie (UML - modele bazowe, statyczne i dynamiczne)	2
Wy5	Jakość oprogramowania, testowanie, zarządzanie zmianami	2
Wy6	Implementacja obiektowa (C++) - poziomy klas	2
Wy7	Implementacja obiektowa (C++) - poziom systemu	2
Wy8	Zaliczenie pisemne	1
		Suma: 15
Forma zajęć – Projekt		Liczba godzin
Proj1	Sprawy organizacyjne, zapoznanie ze środowiskiem MS Visual Studio	2
Proj2	Przetwarzanie sygnałów – generowanie danych wybranego przebiegu oraz losowych zakłóceń, zapis do pliku	2
Proj3	Przetwarzanie sygnałów – wczytywanie danych z pliku, dynamiczna alokacja pamięci na dane jednowymiarowe, programowanie algorytmów filtracji	2
Proj4	Przetwarzanie sygnałów – odporność na błędy, testowanie i debugowanie programu, dokumentacja kodu	2
Proj5	Przetwarzanie obrazów – wczytywanie obrazu z pliku, dynamiczna alokacja pamięci na dane wielowymiarowe	2

Proj6	Przetwarzanie obrazów – struktury danych	2
Proj7	Przetwarzanie obrazów – programowanie wybranych algorytmów przetwarzania danych dwuwymiarowych	2
Proj8	Przetwarzanie obrazów – generowanie obrazów	2
Proj9	Przetwarzanie obrazów – odporność na błędy, organizacja i dokumentacja projektu	2
Proj10	Dynamiczne struktury danych – praca z dynamiczną listą jedno lub dwukierunkową, lub drzewem	2
Proj11	Dynamiczne struktury danych – budowanie dynamicznej struktury danych na podstawie danych zapisanych w pliku	2
Proj12	Dynamiczne struktury danych – wyszukiwanie elementów, zamiana miejscami, usuwanie, sortowanie	2
Proj13	Projekt indywidualny - specyfikacja wymagań, projektowanie	2
Proj14	Projekt indywidualny - implementacja, testowanie	2
Proj15	Projekt indywidualny - odbiory	2
		Suma: 30

STOSOWANE NARZĘDZIA DYDAKTYCZNE

N1. prezentacja multimedialna
N2. praca własna - przygotowanie do projektu
N3. Praca własna - pisanie oraz dokumentowanie programów
N4. Internetowe bazy wiedzy

OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW UCZENIA SIĘ (Wykład)

Oceny (F – formująca (w trakcie semestru), P – podsumowująca (na koniec semestru))	Numer efektu uczenia się	Sposób oceny osiągnięcia efektu uczenia się
F1	PEK_W01 - PEK_W03	zaliczenie pisemne
P = F1		

OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW UCZENIA SIĘ (Projekt)

Oceny (F – formująca (w trakcie semestru), P – podsumowująca (na koniec semestru))	Numer efektu uczenia się	Sposób oceny osiągnięcia efektu uczenia się
--	--------------------------	---

F1	PEK_U01 - PEK_U03 PEK_K01 - PEK_K02	wejściówki, odpowiedzi ustne, sprawozdania (program + dokumentacja)
P = F1		

LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA

LITERATURA PODSTAWOWA

Kernighan B.W., D. M. Ritchie : Język ANSI C

Wirth N.: Algorytmy + Struktury Danych = Programy

LITERATURA UZUPEŁNIAJĄCA

Cohn M., Succeeding with Agile, Addison-Wesley 2010 Weisfeld M., The Object-Oriented Thought Process, Addison-Wesley, 2009

Freeman S., Pryce N., Growing Object-Oriented Software Guided By Tests, Addison-Wesley, 2010

Dathan B., Ramnath S., Object-Oriented Analysis and Design, Springer 2010 B.B

Agarwald, S.P. Tayal, M. Gupta, Software Engineering and Testing, 2010

Prata S.: Szkoła Programowania. Język C++

Stroustrup B.: Język C++

Chomicz P., Uljasz R.: Programowanie w języku C i C++. Poradnik programisty

OPIEKUN PRZEDMIOTU

dr hab. inż. Jacek Reiner tel.: 29-81 email: jacek.reiner@pwr.edu.pl

SUBJECT CARD

Name in Polish: **Informatyka II**

Name in English: **Software Engineering**

Main field of study (if applicable): **Robotics and Process Automation**

Level and form of studies: **I level, full-time**

Kind of subject: **obligatory**

Subject code: **RAM031047**

Group of courses: **no**

	Lecture	Classes	Laboratory	Project	Seminar
Number of hours of organized classes in University (ZZU)	15			30	
Number of hours of total student workload (CNPS)	30			60	
Form of crediting	Crediting with grade			Crediting with grade	
Group of courses					
Number of ECTS points	1			2	
including number of ECTS points for practical (P) classes				2	
including number of ECTS points for direct teacher-student contact (BK) classes	0.6			1.4	

PREREQUISITES RELATING TO KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. Has a basic knowledge of computer algorithms
2. Knows the semantics and syntax of C

SUBJECT OBJECTIVES

- C1. Explain the functions, methods and tools (UML) of software engineering, introduce the object-oriented modeling. Prepare for practical classes with object-oriented programming
- C2. Learn the ability to use procedural programming paradigm
- C3. Learn skills for development of data processing programs

SUBJECT LEARNING OUTCOMES

I. Relating to knowledge:

PEK_W01 - Understands the essence of methodical troubleshooting and application development tools

PEK_W02 - Has knowledge of object-oriented programming paradigm and UML drawing

PEK_W03 - Can read simple programs in C ++

II. Relating to skills:

PEK_U01 - Can apply the procedural programming paradigm, ie. to break down the problem into a set of programming functions through various tasks in C

PEK_U02 - Can apply C language to the processing of complex data sets one and two-dimensional using dynamic data structures

PEK_U03 - Can test and debug programs written in C, use the C language library documentation, knows and applies the principles of proper programming style

III. Relating to social competences:

PEK_K01 - an search and critically analyze information

PEK_K02 - IS able to plan programming task, carried out in accordance with the functional requirements and the reviews

PROGRAM CONTENT

Form of classes – Lecture		Number of hours
Lec1	Introduction to object-oriented programming, examples, tools, CASE	2
Lec2	Models of software development (life cycle)	2
Lec3	Object-oriented requirements analysis (UML - Use case diagrams, activity)	2
Lec4	Object-oriented design (UML - basic models, static and dynamic)	2
Lec5	Software quality, testing, change management	2
Lec6	Implementation of object-oriented (C ++)- classes level	2
Lec7	Implementation of object-oriented (C ++)- system level	2
Lec8	Written assessment	1
		Total hours: 15
Form of classes – Project		Number of hours
Proj1	Organizational issues, introduction into MS Visual Studio environment	2
Proj2	Signal Processing - selected waveform data generation and random failures, write to the file	2
Proj3	Signal Processing - reading a file, dynamic memory allocation of dimensional data, filtering algorithms programming	2
Proj4	Signal processing - robustness, test and debugging the program, code documentation	2

Proj5	Image processing - image reading from a file, dynamic memory allocation for multi-dimensional data	2
Proj6	Image processing - data structures	2
Proj7	Image processing - programming of selected two-dimensional data processing algorithms	2
Proj8	Image processing - image generating	2
Proj9	Image processing - fault tolerance, organization and project documentation	2
Proj10	Dynamic data structures - work with a dynamic list of one or two-way, or a tree	2
Proj11	Dynamic data structure - build a dynamic data structure based on the data stored in the file	2
Proj12	Dynamic data structures - search for items, swap, delete, sort	2
Proj13	Individual project - requirements specification, design	2
Proj14	Individual project - implementation, testing	2
Proj15	Individual project - acceptance	2
		Total hours: 30

TEACHING TOOLS USED		
N1. multimedia presentation N2. self study - preparation for project class N3. Self working - implementation and documenting programs N4. Online Knowledge Base		

EVALUATION OF SUBJECT LEARNING OUTCOMES ACHIEVEMENT (Lecture)		
Evaluation (F – forming (during semester), P – concluding (at semester end))	Learning outcomes number	Way of evaluating learning outcomes achievement
F1	PEK_W01 - PEK_W03	written assessment
P = F1		

EVALUATION OF SUBJECT LEARNING OUTCOMES ACHIEVEMENT (Project)		
Evaluation (F – forming (during semester), P – concluding (at semester end))	Learning outcomes number	Way of evaluating learning outcomes achievement

F1	PEK_U01 - PEK_U03 PEK_K01 - PEK_K02	pre-entry tests, oral answers, statements (source code + documentation)
P = F1		

PRIMARY AND SECONDARY LITERATURE

PRIMARY LITERATURE

SECONDARY LITERATURE

Cohn M., Succeeding with Agile, Addison-Wesley 2010 Weisfeld M., The Object-Oriented Thought Process, Addison-Wesley, 2009

Freeman S., Pryce N., Growing Object-Oriented Software Guided By Tests, Addison-Wesley, 2010

Dathan B., Ramnath S., Object-Oriented Analysis and Design, Springer 2010 B.B

Agarwal, S.P. Tayal, M. Gupta, Software Engineering and Testing, 2010

SUBJECT SUPERVISOR

dr hab. inż. Jacek Reiner tel.: 29-81 email: jacek.reiner@pwr.edu.pl