



**Akademia Nauk Stosowanych**  
**im. Hipolita Cegielskiego w Gnieźnie Uczelnia Państwowa**

**SYLABUS**

<b>Pozycja przedmiotu w planie:</b>		R.I/S.II - 2
<b>1. OGÓLNY OPIS PRZEDMIOTU</b>		
1	Nazwa modułu	Moduł zajęć kierunkowych
2	Nazwa przedmiotu	Programowanie obiektowe
3	Kierunek studiów	Informatyka
4	Poziom studiów	pierwszy
5	Forma studiów	niestacjonarne
6	Profil studiów	praktyczny
7	Rok studiów	pierwszy
8	Semestr przedmiotu	drugi
9	Jednostka prowadząca kierunek studiów	Instytut Nauk Technicznych
10	Liczba punktów ECTS	3
11	Sposób zaliczenia:	Wykład: zaliczenie z oceną, Laboratorium: zaliczenie z oceną
12	Imię i nazwisko nauczyciela (li) akademickiego (ich), stopień lub tytuł naukowy, adres e-mail	Maciej Sobieraj Dr inż. maciej.sobieraj@put.poznan.pl
13	Imię i nazwisko koordynatora(ów) przedmiotu, stopień lub tytuł naukowy, adres e-mail	Maciej Sobieraj Dr inż. maciej.sobieraj@put.poznan.pl
14	Język wykładowy	polski
15	Tryb prowadzenia zajęć	-
16	Sposób prowadzenia zajęć	synchroniczny
17	Narzędzia informatyczne wykorzystywane do prowadzenia zajęć, udostępniania materiałów i komunikacji ze studentami	Platforma Moodle
15	Przedmioty wprowadzające	-
16	Wymagania wstępne	1. Podstawowa znajomość obsługi komputera z systemem operacyjnym Windows 2. Znajomość podstaw programowania
17	<b>Cele przedmiotu:</b>	
C1	Poznanie metodologii programowania obiektowego.	
C2	Nabywanie praktycznej umiejętności projektowania i implementacji, uruchomienia i testowania programów za pomocą języka programowania obiektowego C++.	
C3	Zapoznanie się z podstawami programowania obiektowego niezbędnymi dla każdego informatyka..	
18	Forma zajęć, liczba godzin wymagająca bezpośredniego udziału nauczyciela akademickiego, liczba godzin nakładu pracy studenta	

Forma zajęć		Liczba godzin
1. Wykład		8
2. Laboratoria		24
3.		
Suma godzin		32
lp.	Całkowity nakład pracy studenta	
1.	Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi:	<b>Godzinowe obciążenie studenta</b>
	<b>Wykład: 8</b>	<b>32godzin</b>
	<b>Laboratorium: 24</b>	
Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi 60 godzin, co odpowiada 1 punktowi ECTS.		
2	Bilans nakładu pracy studenta: 1. Przygotowanie do laboratorium: 32 2. Samodzielne studiowanie literatury: 8 3. Przygotowanie do zaliczenia: 8  Łączny nakład pracy studenta wynosi 48 godzin, co odpowiada 2 punktom ECTS.	48 godzin
3	<b>Łączny nakład pracy studenta (pozycja 1+2)</b>	80 godzin
4	<b>Punkty ECTS za przedmiot</b>	3 ECTS
5	Liczba punktów ECTS, którą student musi osiągnąć w ramach zajęć o charakterze praktycznym w tym zajęć laboratoryjnych, warsztatowych, projektowych	2 ECTS
Efekty uczenia się - wiedza	K_W04: ma wiedzę w zakresie znajomości wybranych paradygmatów i języków programowania, podstawowych konstrukcji programistycznych oraz implementacji algorytmów, zna rodzaje i zasady programowania, w tym za pomocą języków programowania wyższego poziomu. K_W05: ma ogólną wiedzę i z zakresu teorii i metod formułowania, konstruowania i stosowania algorytmów; zna typowe algorytmy grafowe, sortowania i wyszukiwania, ma uporządkowaną wiedzę w zakresie znajomości abstrakcyjnych struktur danych oraz ich implementacji.	
Efekty uczenia się - umiejętności	K_U13: potrafi pisać, uruchamiać, śledzić i testować programy w wybranym środowisku programistycznym wykorzystując znajomość paradygmatów i metod programowania. K_U14: ma umiejętności: implementowania algorytmów, konstruowania algorytmów z wykorzystaniem podstawowych technik algorytmicznych, analizy złożoności algorytmów.	
Efekty uczenia się – kompetencje społeczne	K_K01: rozumie potrzebę i zna możliwości ciągłego dokształcania się, krytycznie odnosi się do posiadanej wiedzy, podnoszenia kompetencji zawodowych, osobistych i społecznych. K_K02: ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera-informatyka, w tym jej wpływ na środowisko i	

	związaną z tym odpowiedzialność za podejmowane decyzje; dba o dobre tradycje zawodu informatyka.
--	--------------------------------------------------------------------------------------------------

2. TREŚCI PROGRAMOWE ODNIESIONE DO EFEKTÓW UCZENIA SIĘ		
Forma zajęć	Treści programowe	liczba godzin
<b>Forma: wykład</b>		
1	Wprowadzenie do programowania - Powtórzenie podstaw z języka C++ (zmienne, operatory, instrukcje warunkowe)	2
2	Dostęp do różnego rodzaju danych - Tablice i struktury - Wskaźniki - Funkcje - Konwersje - Operacje na zmiennych typu string	2
3	Definiowanie klas: składowe klasy, modyfikatory dostępu, konstruktory, destruktory - Klasy - Konstruktory i destruktory - Funkcje statyczne	2
4	Dziedziczenie - Definiowanie hierarchii klas - Klasy, dziedziczenie i zgodność typów - Polimorfizm i metody wirtualne - Obiekty jako parametry i dynamiczne rzutowanie - Różne dodatki - Słowo kluczowe const - Przyjaźń w świecie "C++"	2
<b>Forma: laboratorium</b>		
1	Wprowadzenie do programowania - Powtórzenie podstaw z języka C++ (zmienne, operatory, instrukcje warunkowe)	6
2	Dostęp do różnego rodzaju danych - Tablice i struktury - Wskaźniki - Funkcje - Konwersje - Operacje na zmiennych typu string	6
3	Definiowanie klas: składowe klasy, modyfikatory dostępu, konstruktory, destruktory - Klasy - Konstruktory i destruktory - Funkcje statyczne	6
4	Dziedziczenie - Definiowanie hierarchii klas - Klasy, dziedziczenie i zgodność typów - Polimorfizm i metody wirtualne - Obiekty jako parametry i dynamiczne rzutowanie - Różne dodatki - Słowo kluczowe const - Przyjaźń w świecie "C++"	6

3. LITERATURA	
Literatura podstawowa	S. Prata, Język C++. Szkoła programowania. Helion 2012.
Literatura uzupełniająca	M. J. Kubiak, C++. Zadania z programowania z przykładowymi rozwiązaniami, Helion, 2011

4. METODY DYDAKTYCZNE	
<b>Forma</b>	<b>Metody dydaktyczne</b>
<b>Wykład</b>	Metody podające - wykład informacyjny, wykład konwersatoryjny, Metody eksponujące - prezentacja multimedialna
<b>Laboratorium</b>	Metody poszukujące - metody ćwiczeniowo- praktyczne – projekt

5. METODY I KRYTERIA OCENIANIA	
<b>Forma zajęć: wykład</b>	<b>Forma zaliczenia: zaliczenie z oceną</b>
Uzyskane punkty są przeliczane na oceny według następującej skali: Procent punktów                      Ocena 91-100%                                Bardzo dobry 85-90%                                  Dobry plus 76-84%                                  Dobry 66-75%                                  Dostateczny plus 51-65%                                  Dostateczny 0-50%                                    Niedostateczny	
Opis: Test z pytaniami testowymi z jedną poprawną odpowiedzią spośród 4 do wyboru. Test składa się z 10-20 pytań.	
<b>Forma zajęć: laboratorium</b>	<b>Forma zaliczenia: zaliczenie z oceną</b>
Uzyskane punkty są przeliczane na oceny według następującej skali: Procent punktów                      Ocena 91-100%                                Bardzo dobry 85-90%                                  Dobry plus 76-84%                                  Dobry 66-75%                                  Dostateczny plus 51-65%                                  Dostateczny 0-50%                                    Niedostateczny	
Opis: Projekt do wykonania na ostatnich zajęciach. Program komputerowy napisany w języku programowania C++.	
Warunkiem zaliczenia przedmiotu jest uzyskanie oceny pozytywnej ze wszystkich form zajęć.	

	Zatwierdzenie karty opisu zajęć	
	Stanowisko	Podpis
	Tytuł/stopień naukowy, imię nazwisko	
<b>Opracował</b>	Dr inż. Maciej Sobieraj	
<b>Zatwierdził</b>	Dyrektor Instytutu.....	