

## Szczegółowy opis zajęć (KARTA PRZEDMIOTU)

**Nazwa zajęć:** Programowanie komputerów

**Kod zajęć:** 12

**Przynależność do grupy zajęć:** przedmioty wspólne

**Rodzaj zajęć:** kierunkowy, obowiązkowy

**Kierunek studiów:** Informatyka (RAu)

**Poziom studiów:** studia pierwszego stopnia

**Profil studiów:** praktyczny

**Forma studiów:** stacjonarne

**Specjalność (specjalizacja):** -

**Rok studiów:** I, II

**Semestr studiów:** II, III, IV

**Formy prowadzenia zajęć, wraz z liczbą godzin dydaktycznych:**

wykłady – 60 (sem. II – 30, sem. III – 30)

laboratorium – 45 (sem. II – 15, sem. III – 15, IV - 15)

projekt – 15 (sem. IV)

**Język/i, w którym/ch prowadzone są zajęcia:** język polski

**Liczba punktów ECTS:** 10 (sem. II – 4, sem. III – 3, IV - 3)

### 1. Założenia przedmiotu:

Przedmiot ma przekazać wiedzę niezbędną do zrozumienia projektowania i implementowania programów oraz umiejętność implementacji oprogramowania na przykładzie języków C, C++ i C#. Celem przedmiotu jest stworzenie mocnych podstaw zarówno teoretycznych, jak i praktycznych w zakresie tworzenia oprogramowania strukturalnego, obiektowego i generycznego oraz metaprogramowania. Laboratorium oraz projekt umożliwiają zastosowanie w praktyce wiadomości nabytych podczas wykładów, w tym wiedzy koniecznej do tworzenia oprogramowania, w szczególności dotyczącej podstawowych i zaawansowanych technik obiektowych na przykładzie języka C++ i elementów języka C# (analiza, projektowanie i programowanie obiektowe).

### 2. Odniesienie kierunkowych efektów uczenia się do form prowadzenia zajęć oraz sposobów weryfikacji i oceny efektów uczenia się osiągniętych przez studenta:

symbol	zakładane efekty uczenia się <i>student, który zaliczył zajęcia:</i>	formy prowadzenia zajęć	sposoby weryfikacji i oceny efektu uczenia się
Wiedza: zna i rozumie			
K1A_W09 K1A_W11 K1A_W15	Szczegółowe zagadnienia z zakresu programowania strukturalnego i obiektowego	Wykład	Egzamin (sem. IV)
Umiejętności: potrafi			
K1A_U23 K1A_U26	zastosować informacje nt. wybranego zagadnienia z zakresu programowania komputerów	Laboratorium	Program komputerowy (sem. II)
K1A_U23 K1A_U26	rozwiązywać problemy z użyciem programowania strukturalnego	Laboratorium	Program komputerowy (sem. II)
K1A_U07 K1A_U18 K1A_U23	zastosować podstawowe i średniozaawansowane techniki programowania strukturalnego	Wykład, laboratorium	Sprawdzian pisemny (sem. II)
K1A_U23 K1A_U26	zastosować w praktyce informacje nt. wybranego zagadnienia z zakresu programowania komputerów	Laboratorium	Program komputerowy (sem. III)
K1A_U23 K1A_U26	rozwiązywać problemy z użyciem programowania obiektowego	Laboratorium	Program komputerowy (sem. III)
K1A_U23 K1A_U26	zastosować podstawowe i średniozaawansowane techniki programowania obiektowego	Wykład, laboratorium	Sprawdzian pisemny (sem. III)
K1A_U07 K1A_U18 K1A_U23	zgiębić i zaprezentować informacje nt. wybranego zagadnienia z zakresu programowania obiektowego	Laboratorium	Referat (sem. IV)
K1A_U23	zastosować w praktyce informacje nt. wybranego	Laboratorium	Program

K1A_U26	zagadnienia z zakresu zaawansowanego programowania obiektowego		komputerowy (sem. IV)
K1A_U23 K1A_U26	wytwarzać oprogramowanie o dużej złożoności z wykorzystaniem dostępnej dokumentacji technicznej	Projekt	Program komputerowy (sem. IV)

3. Treści programowe zapewniające uzyskanie efektów uczenia się (zgodnie z programem studiów):

Do treści programowych przedmiotu należy nabycie zaawansowanych umiejętności formułowania algorytmów oraz ich implementacji z użyciem technik strukturalnych i obiektowych z elementami programowania generycznego w języku C/C++ w celu rozwiązania prostego zadania inżynierskiego z dziedziny Informatyki.

4. Opis sposobu wyznaczania punktów ECTS:

Forma aktywności	Liczba godzin / punktów ECTS
Liczba godzin zajęć, niezależnie od formy ich prowadzenia	120
Praca własna studenta: analiza treści wykładowych	30
Praca własna studenta: wykonanie zadań laboratoryjnych i przygotowanie sprawozdań	20
Praca własna studenta: zgłębienie wybranego zagadnienia z zakresu programowania obiektowego, opracowanie prezentacji/referatu	10
Praca własna studenta: wykonanie projektów wraz z dokumentacją	90
Inne: przygotowanie do sprawdzianów i egzaminu	30
<b>Suma godzin</b>	<b>300</b>
<b>Liczba punktów ECTS przypisana do zajęć</b>	<b>10</b>

5. Wskaźniki sumaryczne:

- liczba godzin zajęć oraz liczba punktów ECTS na zajęciach z bezpośrednim udziałem nauczycieli akademickich lub innych osób prowadzących zajęcia i studentów: 120 / 10 ECTS
- liczba godzin zajęć oraz liczba punktów ECTS na zajęciach związanych z prowadzoną w Politechnice Śląskiej działalnością naukową w dyscyplinie lub dyscyplinach, do których przyporządkowany jest kierunek studiów – w przypadku studiów o profilu ogólnoakademickim: 120 / 10 ECTS
- liczba godzin zajęć oraz liczba punktów ECTS na zajęciach kształtujących umiejętności praktyczne – w przypadku studiów o profilu praktycznym: 60 / 2 ECTS
- liczba godzin zajęć prowadzonych przez nauczycieli akademickich zatrudnionych w Politechnice Śląskiej jako podstawowym miejscu pracy: 120 (liczba godzin na przedmiot)

6. Osoby prowadzące poszczególne formy zajęć (imię, nazwisko, stopień naukowy lub stopień w zakresie sztuki, tytuł profesora, służbowy adres e-mail):

- 1) Sebastian Deorowicz, prof. dr hab. inż., sebastian.deorowicz@polsl.pl
- 2) Roman Starosolski, dr hab. inż. prof. Pol. Śl., roman.starosolski@polsl.pl
- 3) Dariusz Myszor, dr. inż., dariusz.myszor@polsl.pl

7. Szczegółowy opis form prowadzenia zajęć:

1) wykłady:

- szczegółowe treści programowe:

Wprowadzenie do C++, elementy nieobiektywne C++. Techniki obiektowe na przykładzie języka C++. Paradygmat programowania obiektowego, abstrakcyjne typy danych. Klasy, obiekty, cykl życia obiektów. Operatory przeciążone. Dziedziczenie. Polimorfizm. Informacja o typach czasu wykonania (RTTI). Klasy abstrakcyjne. Dziedziczenie wielobazowe. Mechanizm wyjątków. Wzorce (szablony). Biblioteka języka C++, biblioteka standardowa C++, biblioteka wejścia/wyjścia. Programowanie generyczne; biblioteka STL, kontenery, iteratory i algorytmy. Inteligentne wskaźniki. Metaprogramowanie. Równoległość. Wybrane i zaawansowane elementy biblioteki języka. Wprowadzenie do języka C#, podstawowe typy, dynamiczne zarządzanie pamięcią, opis technologii, mechanizm refleksji, operatory, dziedziczenie, typy generyczne.

- stosowane metody kształcenia, w tym metody i techniki kształcenia na odległość:

*prezentacja multimedialna, samodzielne zgłębienie i opracowanie prezentacji dotyczącej wybranego zagadnienia z zakresu programowania obiektowego, realizacja ćwiczeń tematycznych, realizacja większych projektów programistycznych wraz z konsultacją, prezentacją i dyskusją postępów pracy i opracowaniem dokumentacji*

- forma i kryteria zaliczenia, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:

*Laboratorium, projekt i wykład: w danym semestrze uzyskanie ponad połowy możliwych do uzyskania punktów za wyniki w każdej z kategorii: zadania tematyczne, projekt programistyczny, sprawdzian (sem. 2 i 3), referat (sem. 4).*

*Egzamin: dopuszczenie do egzaminu pod warunkiem zaliczenia laboratorium i projektu w sem. 4, zwolnienie na podstawie ocen uzyskanych w sem. 2-4 (wszystkie muszą być przynajmniej 4.0, a zaliczenie w sem. 4 uzyskane przed rozpoczęciem sesji egzaminacyjnej)*

- organizacja zajęć oraz zasady udziału w zajęciach, ze wskazaniem czy obecność studenta na zajęciach jest obowiązkowa,

*Obecność na zajęciach laboratoryjnych i projektowych obowiązkowa (dopuszczalna i nieobecność nieusprawiedliwiona). Obowiązkowe przechowywanie aktualnej wersji projektu w repozytorium GitHub.*

2) opis pozostałych form prowadzenia zajęć (tematyka laboratoriów i projektów):

Typy danych, operatory, instrukcje. Typy tablicowe i wskaźnikowe. Biblioteki i funkcje biblioteczne. Praca z plikami. Struktury i unie. Implementacja wybranych rozwiązań (np. maszyna stanów, bufor cykliczny). Możliwości C++ w części nieobiektowej. Klasy, konstruktory i destruktory. Przeciążanie operatorów. Dziedziczenie.

Polimorfizm. Mechanizm wyjątków. RTTI. Wzorce (szablony). Kontenery, iteratory i algorytmy STL. Inteligentne wskaźniki. Wątki (podstawy). Wyrażenia regularne. Realizacja złożonych zadań programistycznych w C++ i C#.

8. Opis sposobu ustalania oceny końcowej (zasady i kryteria przyznawania oceny, a także sposób obliczania oceny w przypadku zajęć, w skład których wchodzi więcej niż jedna forma prowadzenia zajęć, z uwzględnieniem wszystkich form prowadzenia zajęć oraz wszystkich terminów egzaminów i zaliczeń, w tym także poprawkowych):

*Laboratorium, projekt i wykład: na podstawie łącznej liczby punktów uzyskanych za wyniki w kategoriach: zadania tematyczne, projekt programistyczny, sprawdzian (sem. 2 i 3), referat (sem. 4).*

*Ocena końcowa:  $(49\% * \text{ocena\_4\_sem} + 51\% * \text{ocena\_egzamin})$  lub  $\text{ocena\_zwolnienie\_z\_egzaminu}$ .*

9. Sposób i tryb uzupełniania zaległości powstałych wskutek:

- nieobecności studenta na zajęciach – *odrabianie nieobecności na laboratorium – terminy ustalone indywidualnie z prowadzącym zajęcia,*
- różnic w programach studiów osób przenoszących się z innego kierunku studiów, z innej uczelni albo wznawiających studia na Politechnice Śląskiej – *ustalane indywidualnie ze studentem na podstawie karty przedmiotu, zrealizowanych treści kształcenia,*

10. Wymagania wstępne i dodatkowe, z uwzględnieniem sekwencyjności zajęć:

Przedmiot: Podstawy programowania komputerów

11. Zalecana literatura oraz pomoce naukowe:

Literatura podstawowa:

- B. W. Kernighan, D.M.Ritche, Język ANSI C, WNT, Warszawa 2004 (lub nowsze)
- ISO/IEC JTC1/SC22/WG14 International Standard—Programming Languages C (draft C11: <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1539.pdf>)
- B. Stroustrup, Język C++, W-wa, WNT, 2008 (lub nowsze)
- ISO/IEC JTC1/SC22/WG21 International Standard—Programming Languages C++ (draft C++14: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3690.pdf>)
- T. Dimes, C# Programming for Beginners: An Introduction and Step-by-Step Guide to Programming in C#

Literatura uzupełniająca:

- K. N. King, Język C. Nowoczesne programowanie. Helion, 2011.
- R. Reese, Wskaźniki w języku C. Przewodnik. Helion, 2014.
- J. Grębosz, Symfonia C++ Standard, tom I/II, wyd. Edition2000, Kraków, 2009 (lub nowsze)
- J. Grębosz, Pasja C++, tom I/II, wyd. Edition2000, Kraków, 2010 (lub nowsze)
- Nicolai M. Josuttis: C++ Biblioteka standardowa Podręcznik Programisty, Wydanie II, Helion 2014
- B. Stroustrup, Projektowanie i rozwój języka C++.
- Troelsen, Pro C# 5.0 and the .NET 4.5 Framework

12. Opis kompetencji prowadzących zajęcia (np. publikacje, doświadczenie zawodowe, certyfikaty, szkolenia itp. związane z treściami programowymi realizowanymi w ramach zajęć):

prof. dr hab. inż. Sebastian Deorowicz

Około 20-letnie doświadczenie w prowadzeniu zajęć z Programowania Komputerów oraz przedmiotów powiązanych z Programowaniem Komputerów.

Opracowanie licznych publikacji wraz z towarzyszącymi im ogólnodostępnymi implementacjami algorytmów praktycznie wykorzystywanych w dziedzinie kompresji danych oraz Bioinformatyce.

Opracowanie edytora tekstu LaTeX (freeware)

Publikacje i implementacje dostępne na stronach <http://sun.aei.polsl.pl/~sdeor/index.php?page=publications> i [http://sun.aei.polsl.pl/~sdeor/index.php?page=my\\_software](http://sun.aei.polsl.pl/~sdeor/index.php?page=my_software).

Promotorstwo kilkudziesięciu prac dyplomowych oraz kilku doktoratów w dziedzinie Informatyki, których część praktyczna obejmowała między innymi implementacje realizowane z wykorzystaniem rozmaitych narzędzi, paradygmatów i języków programowania.

dr hab. inż. Roman Starosolski prof. Pol. Śl,

Około 20-letnie doświadczenie w prowadzeniu zajęć z Programowania Komputerów oraz przedmiotów powiązanych z Programowaniem Komputerów.

Opracowanie licznych publikacji wraz z towarzyszącymi im ogólnodostępnymi implementacjami algorytmów praktycznie wykorzystywanych w dziedzinie kompresji obrazów oraz przetwarzania obrazów.

Publikacje i implementacje dostępne na stronie <http://sun.aei.polsl.pl/~rstaros/#research>.

Uzyskanie 4 patentów krajowych.

Promotorstwo kilkudziesięciu prac dyplomowych w dziedzinie Informatyki, których część praktyczna obejmowała między innymi implementacje realizowane z wykorzystaniem rozmaitych narzędzi, paradygmatów i języków programowania.

dr inż. Dariusz Myszor

Około 10-letnie doświadczenie w prowadzeniu zajęć z Programowania Komputerów oraz przedmiotów powiązanych z Programowaniem Komputerów.

Implementacja licznych aplikacji przeznaczonych do celów naukowych (różnorodnych symulatorów, aplikacji wykorzystujących metody uczenia maszynowego itp.) opisanych w opublikowanych artykułach.

Około 10-letnie doświadczenie zawodowe w implementacji komercyjnego oprogramowania, doświadczenie w zarządzaniu projektami informatycznymi bazującymi na różnorodnych technologiach.

Promotorstwo kilkudziesięciu prac dyplomowych w dziedzinie Informatyki, których część praktyczna obejmowała między innymi implementacje realizowane z wykorzystaniem rozmaitych narzędzi, paradygmatów i języków programowania.

13. Inne informacje: -