

**SYLABUS**

DOTYCZY CYKLU KSZTAŁCENIA 2020/21 – 2023/24

(skrajne daty)

Rok akademicki 2021/2022

**1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE**

|                                                       |                                                                        |
|-------------------------------------------------------|------------------------------------------------------------------------|
| Nazwa przedmiotu                                      | <b>Programowanie obiektowe (PO)</b>                                    |
| Kod przedmiotu*                                       |                                                                        |
| Nazwa jednostki prowadzącej kierunek                  | Kolegium Nauk Przyrodniczych                                           |
| Nazwa jednostki realizującej przedmiot                | Kolegium Nauk Przyrodniczych                                           |
| Kierunek studiów                                      | Mechatronika                                                           |
| Poziom studiów                                        | studia I stopnia                                                       |
| Profil                                                | praktyczny                                                             |
| Forma studiów                                         | niestacjonarne                                                         |
| Rok i semestr/y studiów                               | rok II, semestr 3                                                      |
| Rodzaj przedmiotu                                     | kierunkowy                                                             |
| Język wykładowy                                       | j. polski                                                              |
| Koordinator                                           | dr hab. inż. Lucyna . LENIOWSKA, prof. UR                              |
| Imię i nazwisko osoby prowadzącej / osób prowadzących | dr hab. inż. Lucyna . LENIOWSKA, prof. UR<br>dr inż. Marcin Grochowina |

\* -opcjonalnie, zgodnie z ustaleniami w Jednostce

**1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS**

| Semestr<br>(nr) | Wykł. | Ćw. | Konw. | Lab. | Sem. | ZProj | Prakt. | Inne<br>(jakie?) | Liczba pkt.<br>ECTS |
|-----------------|-------|-----|-------|------|------|-------|--------|------------------|---------------------|
| 4               | 18    |     |       | 18   |      |       |        |                  | 5                   |

**1.2. Sposób realizacji zajęć**

- ☒ X zajęcia w formie tradycyjnej  
☐ zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość

**1.3 Forma zaliczenia przedmiotu (z toku)** (egzamin, zaliczenie z oceną, zaliczenie bez oceny)

WYKŁAD - ZALICZENIE BEZ OCENY / EGZAMIN

ĆWICZENIA LABORATORYJNE - ZALICZENIE Z OCENĄ

**2. WYMAGANIA WSTĘPNE**

Znajomość zagadnień z przedmiotu Podstawy programowania.

### 3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

#### 3.1 Cele przedmiotu

|    |                                                                                                                                                                  |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| C1 | Kontynuacja i rozwinięcie zagadnień realizowanych w zakresie przedmiotu „Podstawy programowania”. Zapoznanie z obiektowymi technikami programowania.             |
| C2 | Nabycie umiejętności praktycznych w zakresie stosowania złożonych struktur danych oraz opracowania programów w języku C++ z zastosowaniem podejścia obiektowego. |

#### 3.2 Efekty uczenia się dla przedmiotu

| EK (efekt uczenia się) | Treść efektu uczenia się zdefiniowanego dla przedmiotu                                                                                                                                                                                                             | Odniesienie do efektów kierunkowych <sup>1</sup> |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| EK_01                  | Zna i rozumie obiektowe techniki programowania. Definiuje podstawowe pojęcia: abstrakcja, dziedziczenie, hermetyczność, polimorfizm. Posiada wiedzę na temat struktur statycznych i dynamicznych. Definiuje klasy i tworzy ich instancje.                          | K_W09                                            |
| EK_02                  | Potrafi pozyskać wiedzę z dostępnych źródeł w celu stworzenia efektywnie działającego oprogramowania, dokonywać interpretacji, a także wyciągać wnioski oraz formułować i uzasadniać opinie                                                                        | K_U01                                            |
| EK_03                  | Potrafi napisać program z zastosowaniem podejścia obiektowego, wykonać projekt w grupie wraz z jego dokumentacją.                                                                                                                                                  | K_U07                                            |
| EK_04                  | Potrafi stworzyć algorytmy i zaprojektować aplikację w celu rozwiązania postawionego zadania. Stosuje typy obiektowe w programowaniu, stosuje mechanizmy: dziedziczenie, hermetyczność, polimorfizm, stosuje konstruktory, destruktory, przeciążenie metod.        | K_U13                                            |
| EK_05                  | Potrafi posługiwać się językiem angielskim na poziomie B2                                                                                                                                                                                                          | K_U16                                            |
| EK_06                  | Potrafi pracować indywidualnie i w zespole – w tym oszacować czas potrzebny na realizację zleconego zadania oraz opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów.                                                                       | K_U18                                            |
| EK_07                  | Rozumie potrzebę ciągłej aktualizacji wiedzy dotyczącej wiodących narzędzi programistycznych, korzystania z wiedzy wynikającej z opinii bazujących na wiedzy i doświadczeniu ekspertów w przypadkach trudności z samodzielnym rozwiązaniem problemów inżynierskich | K_K02                                            |

<sup>1</sup> W przypadku ścieżki kształcenia prowadzącej do uzyskania kwalifikacji nauczycielskich uwzględnić również efekty uczenia się ze standardów kształcenia przygotowującego do wykonywania zawodu nauczyciela.

### 3.3 Treści programowe

#### A. Problematyka wykładu

| Treści merytoryczne                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Programowanie proceduralne a programowanie obiektowe. Ogólne wiadomości o obiektowej technice programowania. Podstawowe pojęcia. Rozwój obiektowych języków programowania i ich podstawowe cechy. |
| 2. Przegląd typów strukturalnych w języku C++. Tablice statyczne i dynamiczne, dostęp do elementów tablicy, wskaźniki. Zakres widoczności zmiennych.                                                 |
| 3. Referencje. Zastosowania referencji i wskaźników w funkcjach                                                                                                                                      |
| 4. Łańcuchy w stylu C++ , klasa string. Rzutowanie i konwersja typów.                                                                                                                                |
| 5. Przegląd typów strukturalnych w C++ c.d.. Pliki, przetwarzanie i obsługa plików.                                                                                                                  |
| 6. Przegląd typów strukturalnych w C++. Struktury, unie i ich zastosowania.                                                                                                                          |
| 7. Pojęcie klasy i obiektu. Odwołania do obiektów, cechy obiektów, hierarchia klas. Przykłady klas.                                                                                                  |
| 8. Metody. Przekazywanie parametrów, parametry domyślne. Przeciążanie metod. Funkcje i klasy zaprzyjaźnione.                                                                                         |
| 9. Konstruktory i destruktory. Przykłady.                                                                                                                                                            |
| 10. Tożsamość obiektu – this. Hermetyzacja. Dziedziczenie jednokrotne i wielokrotne. Istota dziedziczenia. Zgodność typów przy dziedziczeniu. Polimorfizm w hierarchii klas.                         |
| 11. Pokrywanie metod. Metody wirtualne, dynamiczne i abstrakcyjne. Przeciążanie operatorów. Przykłady.                                                                                               |
| 12. Metaprogramowanie. Szablony funkcji, struktur i klas.                                                                                                                                            |
| 13. Podsumowanie przedmiotu. Zaliczenie.                                                                                                                                                             |

#### B. Problematyka ćwiczeń laboratoryjnych

| Treści merytoryczne                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Zajęcia organizacyjne, zasady pracy, warunki zaliczenia, regulamin pracowni. Zapoznanie ze środowiskiem programistycznym, DevC++/CodeBlock, Pliki składowe projektu C++. Operacje wejścia/wyjścia w języku C++. |
| 2. Operacje z zastosowaniem tablic jedno i dwuwymiarowych. Tablice statyczne i dynamiczne. Wskaźniki w odniesieniu do tablic.                                                                                      |
| 3. Struktury i tablice struktur - wprowadzanie, wyświetlanie i inne operacje na danych.                                                                                                                            |
| 4. Klasy – definicja pól i metod. Wskaźnik this. Hermetyzacja i dziedziczenie. Konstruktor, destruktor. Konstruktor kopiujący.                                                                                     |
| 5. Przeciążenie metod i operatorów.                                                                                                                                                                                |
| 6. Szablony metod i klas.                                                                                                                                                                                          |
| 7. Kolokwium. Zajęcia uzupełniające. Zaliczenie                                                                                                                                                                    |
| 8. <b>Projekt nr 1</b> – realizacja tematu wskazanego przez Prowadzącego (zastosowanie tablic dynamicznych i struktur)                                                                                             |
| 9. Obrona projektów nr 1                                                                                                                                                                                           |
| 10. <b>Projekt nr 2</b> – realizacja tematu wskazanego przez Prowadzącego (zastosowanie klas, i ich mechanizmów: dziedziczenia, polimorfizmu; )                                                                    |
| 11. Obrona projektów nr 2                                                                                                                                                                                          |

### 3.4 Metody dydaktyczne

Wykład – wykłady z prezentacją multimedialną,

Laboratorium – rozwiązywanie zadań, praca w grupach, analiza przykładów, dyskusja.

Projekt - projektowanie aplikacji, praca indywidualna, dyskusja.

#### 4. METODY I KRYTERIA OCENY

##### 4.1 Sposoby weryfikacji efektów uczenia się

| Symbol efektu | Metody oceny efektów uczenia się<br>( np.: kolokwium, egzamin ustny, egzamin pisemny,<br>projekt, sprawozdanie, obserwacja w trakcie zajęć) | Forma zajęć<br>dydaktycznych<br>( w, ćw, ...) |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| EK_01         | Egzamin, kolokwium, sprawozdanie, wykonanie<br>ćwiczeń, zaliczenie sprawdzianów                                                             | w., lab.                                      |
| EK_02         | kolokwium, sprawozdanie, wykonanie ćwiczeń,<br>zaliczenie sprawdzianów                                                                      | lab.                                          |
| EK_03         | kolokwium, sprawozdanie, wykonanie ćwiczeń,<br>zaliczenie sprawdzianów                                                                      | lab.                                          |
| EK_04         | Egzamin, kolokwium, sprawozdanie, wykonanie<br>ćwiczeń, zaliczenie sprawdzianów                                                             | w., lab.                                      |
| EK_05         | Egzamin, kolokwium                                                                                                                          | w.                                            |
| EK_06         | Projekt, obserwacja w trakcie zajęć, kolokwium                                                                                              | lab.                                          |
| EK_07         | Egzamin, obserwacja w trakcie zajęć                                                                                                         | w., lab.                                      |

##### 4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Egzamin - uzyskanie co najmniej 50% punktów z egzaminu.

Punktacja przyjęta podczas oceny egzaminu:

| Ocena z przedmiotu     |            |             |             |             |             |              |
|------------------------|------------|-------------|-------------|-------------|-------------|--------------|
| Przedział<br>punktacji | 0%-<br>50% | 51%-<br>60% | 61%-<br>70% | 71%-<br>80% | 81%-<br>90% | 91%-<br>100% |
| Ocena                  | 2,0        | 3,0         | 3,5         | 4,0         | 4,5         | 5,0          |

**Laboratorium** – ocena z odpowiedzi i/lub sprawdzianów wejściowych ; dwa kolokwia pisemne, dwa projekty, aktywność na zajęciach.

Kolokwium - ocenę pozytywną z kolokwium student uzyskuje w przypadku uzyskania minimum połowy możliwych do uzyskania punktów.

Projekt – ocena wykonanej aplikacji i jej dokumentacji; ocena z odpowiedzi na zadane pytania. Ocenę końcową z projektu oblicza się jako średnią 3 ocen (aplikacji , dokumentacji, odpowiedzi)

Ocenę końcową z laboratorium oblicza się na podstawie średniej ocen otrzymanych z kolokwium, projektów i odpowiedzi / sprawdzianów oraz z aktywności w proporcji 30% ocena z kolokwium, 30% ocena z projektu i 20% ocena aktywności, 20% ocena ze sprawdzianów wejściowych, przy czym wszystkie oceny muszą być pozytywne.

## 5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

| Forma aktywności                                                                                          | Średnia liczba godzin na zrealizowanie aktywności |
|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| Godziny kontaktowe wynikające z harmonogramu studiów                                                      | 36                                                |
| Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)                             | 2 - egzamin<br>5 - konsultacje                    |
| Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.) | 89                                                |
| SUMA GODZIN                                                                                               | 132                                               |
| <b>SUMARYCZNA LICZBA PUNKTÓW ECTS</b>                                                                     | <b>5</b>                                          |

\* Należy uwzględnić, że 1 pkt ECTS odpowiada 25-30 godzin całkowitego nakładu pracy studenta.

## 6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

|                                  |             |
|----------------------------------|-------------|
| wymiar godzinowy                 | Nie dotyczy |
| zasady i formy odbywania praktyk |             |

## 7. LITERATURA

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Literatura podstawowa:</p> <p>G. Grebosz, „Symfonia C++”, Oficyna Kallimach, 1996 t1-3</p> <p>Jerzy Grębosz Opus magnum C++11. Programowanie w języku C++, Helion 2017, t.1-3</p> <p>Jerzy Grębosz Opus magnum C++. Misja w nadprzestrzeń C++14/17. Tom 4. Helion 2020.</p> <p>S. Prata, „Język C++. Szkoła programowania. Wydanie V”, Helion 2014</p> <p>H. Schildt C++. Sztuka programowania, Helion, 2010</p> <p>Andrzej Stasiewicz, C++. Ćwiczenia praktyczne. Wydanie III, Helion 2011</p> <p>Robert C. Martin <b>Mistrz czystego kodu. Kodeks postępowania profesjonalnych programistów</b>, Helion 2015</p> <p><a href="http://www.cplusplus.com/reference/">http://www.cplusplus.com/reference/</a></p> <p><a href="http://en.cppreference.com/w/cpp">http://en.cppreference.com/w/cpp</a></p>                                                                                                                                                                                                             |
| <p>Literatura uzupełniająca:</p> <p>B. Stroustrup, „Programowanie. Teoria i praktyka z wykorzystaniem C++”, Helion 2020</p> <p>Jacek Galowicz <b>C++17 STL. Receptury</b>, Helion 2018</p> <p>G. Grebosz, „Pasja C++”, Oficyna Kallimach, 1996 t1-2</p> <p>A. Stasiewicz, „C++. Ćwiczenia praktyczne”, Helion 2011.</p> <p>S. Oualline, „Jak NIE programować w C++”, Mikom 2003</p> <p>T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein, „Wprowadzenie do algorytmów”, WNT 2004</p> <p><a href="http://4programmers.net/">http://4programmers.net/</a></p> <p>Grochowina M., Leniowska L. Gala-Błądzińska A.: The prototype device for non-invasive diagnosis of arteriovenous fistula condition using machine learning methods, <b>Springer Nature Sci Rep 10</b>, paper No. 72336 (2020). <a href="https://doi.org/10.1038/s41598-020-72336-5">https://doi.org/10.1038/s41598-020-72336-5</a>, <a href="http://www.nature.com/articles/s41598-020-72336-5">www.nature.com/articles/s41598-020-72336-5</a></p> |

Akceptacja Kierownika Jednostki lub osoby upoważnionej