(KARTA PRZEDMIOTU)

Nazwa zajęć: **Programowanie w C/C++**

Kod zajęć: **FT-Isp-35A/2**

Przynależność do grupy zajęć: **pozostałe**

Rodzaj zajęć: **podstawowy / kierunkowy obieralny**

Kierunek studiów: **FIZYKA TECHNICZNA**

Poziom studiów: **studia pierwszego stopnia**

Profil studiów: **praktyczny**

Forma studiów: **stacjonarne**

Specjalność (specjalizacja):

Rok studiów: **1**

Semestr studiów: **2**

Formy prowadzenia zajęć, wraz z liczbą godzin dydaktycznych:

wykłady – 15

laboratorium – 45

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

Liczba punktów ECTS *(zgodnie z programem studiów)*: **5**

* – *pozostawić właściwe*

1. Założenia przedmiotu:

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 *student, który zaliczył zajęcia:*	formy prowadzenia zajęć	sposoby weryfikacji i oceny efektu uczenia się
Wiedza: zna i rozumie			
K1P_W10 K1P_W16	Zna podstawy programowania w C++ oraz związki z językiem C. Zna podstawowe instrukcje, oraz rozwiązania w C i C++ dla operacji we/wy, wskaźników, tablic, łańcuchów i funkcji. Rozumie reguły pisania poprawnych i dobrych programów.	wykład	Ćwiczenia laboratoryjne i indywidualnie realizowane zadania w ramach laboratorium
K1P_W10 K1P_W16	Rozumie zasady programowania obiektowego. Rozumie podstawowe pojęcia: klasa, obiekt, pole, metoda, konstruktor, destruktor.	wykład	Ćwiczenia laboratoryjne i indywidualnie realizowane zadania w ramach laboratorium
K1P_W10 K1P_W16	Rozumie pojęcia otaczania, dziedziczenia, metod wirtualnych, dynamicznej alokacji pamięci, i polimorfizmu. Wie, co to są szablony i kontenery i potrafi je zastosować.	wykład	Ćwiczenia laboratoryjne i indywidualnie realizowane zadania w ramach laboratorium
Umiejętności: potrafi			
K1P_W10 K1P_U08	Potrafi pisać proste programy obiektowe w C++ wykorzystujące dziedziczenie, polimorfizm, przeciążanie operatorów, funkcje zaprzyjaźnione, strumienie, dynamiczną alokację obszarów pamięci.	laboratorium	indywidualne zadanie wykonane w czasie zajęć laboratoryjnych, sprawdzian
K1P_W10 K1P_U08 K1P_U15	Potrafi sformułować algorytm rozwiązania prostego zadania i zapisać go w postaci zorientowanego obiektowo programu w C++.	laboratorium	indywidualne zadanie wykonane w czasie zajęć laboratoryjnych, sprawdzian
K1P_W10 K1P_W16 K1P_U08 K1P_U15	Potrafi opracować proste szablony i kontenery. Umie wykorzystać wybrane elementy biblioteki STL.	laboratorium	indywidualne zadanie wykonane w czasie zajęć laboratoryjnych, sprawdzian
Kompetencje społeczne: jest gotów do			
K1A_K04	współdziałania w grupie wykonującej określone podzadania związane z programowaniem większego zadania wymagającego definicji szeregu klas.	laboratorium	zadanie wykonane w grupie studentów w trakcie zajęć laboratoryjnych

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

Zasadniczym celem zajęć jest przekazanie studentom wiedzy oraz wykształcenie praktycznych umiejętności potrzebnych do analizy problemów inżynierskich oraz tworzenia programów w językach C oraz C++, ze szczególnym uwzględnieniem programowania obiektowego.

4. Opis sposobu wyznaczania punktów ECTS:

Forma aktywności	Liczba godzin / punktów ECTS

Liczba godzin zajęć, niezależnie od formy ich prowadzenia	60 /2
Praca własna studenta: **przygotowanie do laboratoriów**	30 /1
Praca własna studenta: **przygotowanie do sprawdzianów**	30 /1
Inne: **przygotowanie do konsultacji**	5
Suma godzin	125
Liczba punktów ECTS przypisana do zajęć	5

Objaśnienia:
* – praca własna studenta, należy wymienić formy aktywności, np. *przygotowanie do zajęć, interpretacja wyników, opracowanie raportu z zajęć, przygotowanie do egzaminu, zapoznanie się z literaturą, przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania itp.*
** – inne np. *dodatkowe godziny zajęć*

5. Wskaźniki sumaryczne:
   - liczba godzin zajęć oraz liczba punktów ECTS na zajęciach z bezpośrednim udziałem nauczycieli akademickich lub innych osób prowadzących zajęcia i studentów: **60 / 2**
   - 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:
   - liczba godzin zajęć oraz liczba punktów ECTS na zajęciach kształtujących umiejętności praktyczne – w przypadku studiów o profilu praktycznym: **50 / 2**
   - liczba godzin zajęć prowadzonych przez nauczycieli akademickich zatrudnionych w Politechnice Śląskiej, jako podstawowym miejscu pracy: **60**
6. Osoby prowadzące poszczególne formy zajęć *(imię, nazwisko, stopień naukowy lub stopień w zakresie sztuki, tytuł profesora, służbowy adres e-mail)*:

   Wykład:

   dr inż. Adam Pawlak, adiunkt, *adam.pawlak@polsl.pl*

   Laboratoria:

   dr inż. Tomasz Garbolino, adiunkt, *tomasz.garbolino@polsl.pl*

   dr inż. Dariusz Polok, adiunkt, *dariusz.polok.@polsl.pl*

7. Szczegółowy opis form prowadzenia zajęć:
   1) wykłady:

      – szczegółowe treści programowe:

      - **Podstawy programowania w C++: zmienne, typy, instrukcje, łańcuchy, strumienie we/wy.**
      - **Przeciążanie funkcji, argumenty domyślne funkcji, referencje, funkcje ze zmienną liczbą argumentów.**
      - **Idea programowania obiektowego.**
      - **Pojęcie klasy, obiektu, pola i metody.**
      - **Cykl życia obiektów**
      - **Operator zasięgu „::”, wskaźnik „this”.**
      - **Deklaracje i dyrektywy „using”.**
      - **Konstruktory i destruktory. Konstruktor domyślny. Lista inicjalizacyjna.**
      - **Funkcje i klasy zaprzyjaźnione.**
      - **Przeciążanie operatorów.**
      - **Wskaźniki, referencje i dynamiczna alokacja pamięci.**
      - **Dziedziczenie. Polimorfizm. Metody wirtualne. Klasy abstrakcyjne.**
      - **Dziedziczenie wielokrotne i klasy wirtualne.**
      - **Szablony funkcji i klas.**
      - **Wyjątki.**
      - **Wybrane elementy biblioteki standardowej.**

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

      Materiał prezentowany na wykładzie jest wcześniej udostępniane studentom na PZE.

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

      Wobec braku egzaminu przedmiot jest zaliczany na podstawie wyników z laboratorium, tj. 4 sprawdzianów oraz wyników z zadań laboratoryjnych.

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

      Wykłady są realizowane w 4 blokach po 4h. Obecność studenta na wykładzie nie jest obowiązkowa.

   2) Laboratoria:

- szczegółowe treści programowe

- **Przypomnienie najważniejszych zagadnień dotyczących języków C, poznanych w ramach przedmiotu Podstawy programowania.**
- **Operacje związane ze wskaźnikami, wskaźniki do funkcji, tablice wskaźników do funkcji.**
- **Funkcje - przeciążanie, argumenty domyślne funkcji, funkcje ze zmienną liczbą argumentów, przekazywanie argumentów przez wartość, wskaźnik i referencję.**
- **Operator zasięgu „::", przestrzenie nazw, podział programu na moduły.**
- **Programowanie obiektowe. Klasy, obiekty, metody. Konstruktory i destruktory, konstruktory domyślne, listy inicjalizacyjne. Statyczne i dynamiczne tworzenie obiektów.**
- **Kontrola dostępu do składowych. Pola i metody static. Klasy i funkcje zaprzyjaźnione. Wskaźnik this.**
- **Przeciążanie operatorów. Dziedziczenie. Polimorfizm. Metody wirtualne. Klasy abstrakcyjne.**
- **Operacje plikowe. Szablony i kontenery.**

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

Zajęcia na laboratorium polegają na rozwiązywaniu zadań z programowania w językach C i C++. Tylko na 3 pierwszych laboratoriach przypominany i ćwiczony jest język C. Zakłada się bowiem, że wstępne umiejętności z programowania w języku C studenci nabywają w ramach przedmiotu „Podstawy programowania". Przykładowe zadania są udostępniane studentom na PZE.

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

Wobec braku egzaminu przedmiot jest zaliczany na podstawie wyników z laboratorium, tj. 4 sprawdzianów oraz wyników z zadań laboratoryjnych.

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

Zajęcia laboratoryjne są obowiązkowe.

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):

   Wobec braku egzaminu przedmiot jest zaliczany na podstawie wyników z laboratorium, tj. 4 sprawdzianów oraz wyników z zadań laboratoryjnych.

9. Sposób i tryb uzupełniania zaległości powstałych wskutek:
   – nieobecności studenta na zajęciach,

   Tylko usprawiedliwioną w ciągu tygodnia nieobecność na zajęciach można odrobić w terminie wskazanym przez prowadzącego.

   – różnic w programach studiów osób przenoszących się z innego kierunku studiów, z innej uczelni albo wznawiających studia na Politechnice Śląskiej,

   O włączaniu do zajęć lub zwalnianiu z części zajęć studentów przenoszących się w „ramach różnic programowych" decyduje kierownik przedmiotu.

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

    Zakłada się, ze student/tka zaliczył wcześniej przedmiot „Podstawy programowania"

11. Zalecana literatura oraz pomoce naukowe:

    Literatura podstawowa:

S. Prata, „Język C++. Szkoła programowania", Wyd. VI, Helion, 2012
B. Kernighan, D. Ritchie "Jezyk ANSI C. Programowanie", Wyd. II, 2010
J. Grębosz, „Symfonia C++, ISO Standard", Wyd. IIIB, Editions 2000 Kraków, 2010
J. Grębosz, „Pasja C++", Wyd. III, Editions 2000 Kraków, 2003
B. Stroustrup, „Programowanie. Teoria i praktyka z wykorzystaniem C++", Wyd. II poprawione, Helion, 2013
A. Allain, „C++. Przewodnik dla początkujących", Helion, 2014
R. Sokół, „Wstęp do programowania w języku C++", Helion, 2005

    Literatura uzupełniająca:

B. Stroustrup, „Język C++. Kompendium wiedzy", Helion, 2014
S. Rao, „C++. Dla każdego", Wyd. VII, Helion, 2014
N. M. Josuttis, „C++. Biblioteka standardowa. Podręcznik programisty", Wyd. II, Helion, 2014
Z. Koza, „Język C++. Pierwsze starcie", Helion, 2012
A. Stasiewicz, „C++. Ćwiczenia praktyczne", Wyd. III, Helion, 2011

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

Dr inż. Adam Pawlak wykładał przez szereg lat w Instytucie Elektroniki zagadnienia programowania w języku C++, natomiast projektowaniem obiektowym zajmował się już przed 30 laty.

[1] Pawlak A., Jørgensen H., *Holistic Design of Collaborative Networks of Design Engineering Organizations,* 6th IFIP Working Conf. on Virtual Enterprise, PRO-VE 2015, Albi, France, October 5-7, 2015, Risks and Resilience of Collaborative Networks, Springer Int. Publishing, 2015. pp. 612-62.

[2] Pawlak A., Fraś P., Penkala P., *Web services-based collaborative system for distributed engineering*, PRO-VE'08 9th IFIP Working Conference on Virtual Enterprises, Poznan, Poland, 8 - 10 Sept. 2008, in *Pervasive Collaborative Networks*, Edited by Luis M. Camarinha-Matos and Willy Picard, Springer, pp. 463-472.

[3] Wrona W., Pawlak A.: *VLSI Integrated Circuit Design Representation in an Object-oriented CAD Environment*, North-Holland, Microprocessing and Microprogramming, Vol. 32 (1991), pp 85-92.

[4] Papazoglou M, Pawlak A., Wrona W.: *Multiprocessor Modelling System as an Example of Object-Oriented Development*, North-Holland, Microprocessing and Microprogramming, Vol. 25, (1989), pp. 213-220

[5] Pawlak A., Wrona W.: *A Modern Object-Oriented Programming Language as a HDL*, Proc. IFIP Conf. on Computer Hardware Description: Languages and their Applications, Amsterdam, 1987.

dr inż. Tomasz Garbolino

1. **Ukończenie studiów podyplomowych „Sieci Komputerowe i Systemy Mikrokomputerowe" na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej.**
2. **Prowadzanie w Instytucie Elektroniki zajęć dotyczących podstaw programowania komputerów.**
3. **Praktyczne wykorzystanie języka C i C++ do opracowania oprogramowania wspierającego obliczenia naukowe, których wyniki zostały opublikowane m. in. w poniższych pracach:**

[1] T. Garbolino, A. Hławiczka: "A New LFSR with D and T Flip Flops as an Effective Test Pattern Generator for VLSI Circuits", [in:] Proc. of EDCC 3 - Third European Dependable Computing Conference, Prague, Czech Republic, September 15-17, 1999, Eds: J. Hlavicka, E. Maehle, A. Pataricza, Lecture Notes in Computer Science, Vol. 1667, Springer, Berlin, 1999, pp. 321-338.

[2] T. Garbolino, A. Hławiczka: "Efficient Test Pattern Generators Based on Specific Cellular Automata Structures". Microelectronics Reliability, Elsevier Publishing Ltd. Volume 42, Issue 6, June 2002, pp. 975-983.

[3] O. Novak, Z. Pliva, J. Nosek, A. Hławiczka, T. Garbolino, K. Gucwa: "Test-per-clock logic BIST with semi-deterministic test patterns and zero-aliasing compactor". Journal of Electronic Testing: Theory and Applications, vol. 20, no. 1, 2004, pp. 109-122.

[4] G. Papa, T. Garbolino, F. Novak, A. Hławiczka: "Deterministic test pattern generator design with genetic algorithm approach". Journal of Electrical Engineering, vol. 58, no. 3, 2007, pp. 121-127.

[5] T. Garbolino, G. Papa: "Genetic algorithm for test pattern generator design". Automatic evolution of circuits. Applied Intelligence, vol. 32, no. 2, 2010, pp. 193-204.

[6] G. Papa, T. Garbolino: "Stochastic Approach to Test Pattern Generator Design", [in:] Dritsas I. (ed.): Stochastic Optimization - Seeing the Optimal for the Uncertain. InTech, 2011, pp. 75-94.

13. Inne informacje:

Materiały do przedmiotu są umieszczane na PZE pod adresem: https://platforma.polsl.pl/rau3/enrol/index.php?id=252