KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Nazwa przedmiotu

Języki programowania

Kod ECTS

11.3.1569

Nazwa jednostki prowadzącej przedmiot

Instytut Informatyki

Studia

wydział	kierunek		
Wydział Matematyki, Fizyki i Informatyki	Informatyka	poziom	pierwszego stopnia
		forma	stacjonarne
		moduł specjalnościowy	wszystkie
		specjalizacja	wszystkie

Nazwisko osoby prowadzącej (osób prowadzących)

prof. UG, dr hab. Piotr Arłukowicz; mgr Michał Zakrzewski

Formy zajęć, sposób ich realizacji i przypisana im liczba godzin

Formy zajęć

Wykład, Ćw. laboratoryjne

Sposób realizacji zajęć

zajęcia on-line, zajęcia w sali dydaktycznej

Liczba godzin

Ćw. laboratoryjne: 30 godz., Wykład: 30 godz.

Liczba punktów ECTS

7

Przedmiot w wymiarze 30h wykładu, 30h lab. + praca własna studenta

Termin realizacji przedmiotu

2022/2023 letni

Status przedmiotu

obowiązkowy

Język wykładowy

polski

Metody dydaktyczne

- Analiza zdarzeń krytycznych (przypadków)
- Programowanie na żywo (life coding)
- Rozwiązywanie zadań
- Wykonywanie doświadczeń
- Wykład z prezentacją multimedialną

Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne

Sposób zaliczenia

- Zaliczenie na ocenę
- Egzamin

Formy zaliczenia

- egzamin pisemny z pytaniami (zadaniami) otwartymi
- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru

Podstawowe kryteria oceny

Ocenie podlega rozumienie zasad poprawnego programowania oraz stopień znajomości języka wykładowego oraz ewentualnie pozostałych z listy wyboru.
Przedmiot kończy się egzaminem pisemnym w formie testu, z którego nalezy uzyskać min 50%.
Aby przystąpic do egzaminu, trzeba zaliczyć ćwiczenia laboratoryjne - również na min 50%Na ćwiczeniach laboratoryjnych zdobywa się oceny cząstkowe na podstawie zaimplementowanych programów komputerowych i innych aktywności.

Sposób weryfikacji założonych efektów uczenia się

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja postawy studenta
	Wiedza						
K_W05	x	x					
P_W01	x	x					
P_W02	x	x					
P_W03	x	x					
	Umiejętności						
K_U06			x	x	x		x
K_U09				x	x	x	x
P_U01		x		x			
P_U02		x		x			
P_U03		x		x			
	Kompetencje						
K_K01						x	x
K_K03						x	x
P_K01						x	x
P_K02						x	x

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Ukończony 'Wstęp do programowania'

B. Wymagania wstępne

Rozumienie podstawowych konstrukcji programistycznych, takich jak instrukcje warunkowe, pętle, funkcje, rekurencja
Znajomość języka C w stopniu średniozaawansowanym (podstawowe typy danych, tablice, struktury, wskaźniki)
Znajomość podstawowych zasad poprawnego programowania strukturalnego i dobrych praktyk programistycznych

Cele kształcenia

Celem przedmiotu jest zapoznanie słuchaczy z zasadami programowania strukturalnego i obiektowego w połączeniu z dobrymi praktykami programistycznymi na bazie wybranych języków programowania: C/C++, GoLang, Rust, Lua, Perl.

Treści programowe

Kurs obejmuje wybrany język z grupy C/C++, Rust, GoLang, Perl, Lua. Celowo unikamy języków typowo obiektowych, ponieważ będą one na przedmiocie "Wstęp do programowania obiektowego" lub "Programowanie Obiektowe".
W trakcie kursu przedstawiona zostanie mechanika programowania we wszystkich językach przykładowych. Udostępnione porównanie pozwoli lepiej zrozumieć jak programuje się w ogólnym zarysie. Następnie uwaga zostanie skupiona na jednym z języków, aby poznać go dokładniej.
Kurs zacznie się od powtórzenia programowania w C (ponieważ taki język studenci już mieli wyjaśniony wcześniej).
Przejdziemy następnie do podstawowych konstrukcji sterujących: wyrażeń warnukowych, pętli, i innych elementów, w każdym z języków przykładowych.
Po ogólnym wstępie skupimy uwagę na wskaźnikach w C i dynamicznych strukturach danych, a potem przejdziemy do średniozaawansowanego kursu GoLang. Możliwe będzie także dodatkowe studiowanie pozostałych języków z listy.

Wykaz literatury

1. C++:
   - Thinking in C++, Vol. 1: Introduction to Standard C++, 2nd Edition, ISBN-13: 978-0139798092
   - Thinking in C++, Volume 2: Practical Programming, ISBN-13: 978-0130353139
   - The C++ Programming Language, 4th Edition, ISBN-13: 978-0321563842
   - Karol Kuczmarski, *Kurs C++ (Od zera do kier kodera)*, tutorial (wydanie polskie - na Licencji GNU Wolnej Dokumentacji): http://www.cs.put.poznan.pl/arybarczyk/Kurs%20C++.pdf
2. Rust:
   - The Rust Programming Language (Covers Rust 2018), ISBN-13 : 978-1718500440
   - Programming Rust: Fast, Safe Systems Development, ISBN-13: 978-1491927281
3. Perl:
   - Programming Perl (3rd Edition), ISBN-13: 978-0596000271
   - Perl Cookbook, Second Edition, ISBN-13 : 978-0596003135

4. LUA:
   - Programming in Lua, fourth edition, ISBN-13 : 978-8590379867
   - Lua Programming: Syntax, Concepts, and Examples - 3nd Edition, ISBN-13 : 979-8589981629
5. GO:
   - Go Programming Language, The (Addison-Wesley Professional Computing Series) , ISBN-13: 978-0134190440
   - Head First Go, ISBN-13: 978-1491969557
   - Hands-On Software Engineering with Golang: Move beyond basic programming to design and build reliable software with clean code, ISBN-13: 978-1838554491

Dostępne jest wiele więcej książek o programowaniu w tych językach.

Kierunkowe efekty uczenia się

K_W05: ma ogólną wiedzę na temat różnych paradygmatów programowania i języków programowania; szczegółowo zna metody i wzorce projektowania i programowania obiektowego
K_U06: potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców
K_U09: potrafi oceniać przydatność paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu
K_K01: zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego uczenia się
K_K03: potrafi i jest gotów formułować opinie na temat podstawowych zagadnień informatycznych

Wiedza

P_W01: student wyjaśnia działanie podstawowych konstrukcji programistycznych takich jak instrukcje wyboru, pętle, bloki, struktury itp.
P_W02: student rozumie na czym polega praca z danymi umieszczonymi w pamięci komputera oraz na dysku.
P_W03: student rozróżnia różne rodzaje języków programowania i potrafi wskazać ich najważniejsze zastosowania.

Umiejętności

P_U01: student potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców
P_U02: student potrafi rozwiązań proste problemy programistyczne przy użyciu metod programowania strukturalnego i innych.
P_U03: student potrafi zastosować poznane konstrukcje programistyczne do pisania działających i poprawnych programów.

Kompetencje społeczne (postawy)

P_K01: zna ograniczenia własnej wiedzy w dziedzinie programowania i rozumie potrzebę dalszego uczenia się i dokładniejszego poznawania języków programowania
P_K02: potrafi i jest gotów formułować opinie na temat podstawowych zagadnień informatycznych związanych z poznanymi językami programowania

Kontakt

piotr.arlukowicz@ug.edu.pl