



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Wprowadzenie do informatyki [S1Inf1>WdI]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/1

Studia w zakresie (specjalność)
–

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obieralny

Liczba godzin

Wykład
30

Laboratorium
16

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

4,00

Koordynatorzy

prof. dr hab. inż. Jerzy Nawrocki
jerzy.nawrocki@put.poznan.pl

Wykładowcy

Wymagania wstępne

Rozpoczynając ten przedmiot, student powinien mieć wiedzę i umiejętności określone w rozporządzeniu Ministra Edukacji Narodowej z dnia 30 stycznia 2018 r. w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia (<https://isap.sejm.gov.pl/isap.nsf/download.xsp/WDU20180000467/O/D20180467.pdf>): * matematyka: zakres rozszerzony, * informatyka: zakres podstawowy.

Cel przedmiotu

Przedmiot ma pomóc studentom w zrozumieniu, czym jest informatyka, poprzez zapoznanie ich z podstawowymi obszarami informatyki i ukazanie relacji między nimi. Ponadto przedmiot wspiera studentów w nabyciu podstawowych umiejętności programistycznych.

Przedmiotowe efekty uczenia się

Wiedza

1. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie kluczowych zagadnień informatyki, oraz wiedzę szczegółową w zakresie wybranych zagadnień tej dyscypliny nauki
2. ma wiedzę o istotnych kierunkach rozwoju i najważniejszych osiągnięciach informatyki oraz innych

pokrewnych dyscyplin naukowych, w szczególności elektroniki, telekomunikacji oraz automatyki i robotyki

3. ma podstawową wiedzę o cyklu życia systemów informatycznych, zarówno sprzętowych jak i programowych, a w szczególności o zachodzących w nich kluczowych procesach

4. ma wiedzę nt. kodeksów etycznych dotyczących informatyki, jest świadomy zagrożeń związanych z przestępczością elektroniczną, oraz rozumie specyfikę systemów krytycznych ze względów bezpieczeństwa (ang. mission-critical systems)

5. ma podstawową wiedzę nt. patentów, ustawy prawo autorskie i prawa pokrewne oraz ustawy o ochronie danych osobowych oraz transferu technologii w szczególności w odniesieniu do rozwiązań informatycznych

Umiejętności

1. potrafi odpowiednio posługiwać się technikami informacyjno-komunikacyjnymi, znajdującymi zastosowanie na różnych etapach realizacji przedsięwzięć informatycznych

2. potrafi dostrzec w procesie formułowania i rozwiązywania zadań informatycznych również aspekty pozainformatyczne, w szczególności kwestie społeczne, prawne i ekonomiczne

3. potrafi organizować, współdziałać i pracować w grupie, przyjmując w niej różne role oraz potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania

Kompetencje społeczne

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe

2. ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich oraz zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- bieżącą ocenę w formie krótkich quizów;
- test śródsesestralny;
- ocenę pracy zespołowej;
- egzamin.

Treści programowe

Przedmiot obejmuje następujące treści:

Programowanie imperatywne

Układy cyfrowe

Komputery i język asemblera

Podprogramy

Przetwarzanie tekstu

Programowanie obiektowe

Metody numeryczne

Złożoność obliczeniowa

Bazy danych i uczenie maszynowe

Obliczenia współbieżne

Sieci komputerowe i kryptografia

Inżyniera oprogramowania

Profesjonalizm w informatyce

Tematyka zajęć

Tematyka wykładów i skorelowanych z nimi ćwiczeń laboratoryjnych obejmuje następujące zagadnienia:

- * Porównanie języków C i Python: proste programy, deklarowanie i przetwarzanie zmiennych prostych, wczytywanie i drukowanie zmiennych, instrukcja warunkowa, instrukcje pętli, tablice, rekordy, koncepcja modularyzacji, funkcje;
- * Podstawowe układy cyfrowe: algebra Boole'a, bramki logiczne, sumatory, dekodery i multipleksery, przerzutniki, rejestry, cyfrowe układy biochemiczne;
- * Budowa prostego komputera: architektura von Neumanna, mikroprocesory a mikrokontrolery, maszyna Turinga;
- * Języka assemblera NASM: rozkazy dot. liczb całkowitych, proste programy, system szesnastkowy, liczby

całkowite, reprezentacja liczb ujemnych, rozkazy skoku i ich implementacja;

* Podprogramy w języku C i Python: przekazywanie parametrów, funkcja jako parametr, implementacja podprogramów w języku assemblera;

* Przetwarzanie tekstu w języku C i Python: reprezentacja łańcuchów znaków, wczytywanie, przetwarzanie i drukowanie tekstu, elementy standardowej biblioteki wejścia/wyjścia, wyrażenia regularne, język AWK;

* Podstawy programowania obiektowego w języku C++ i Python: rekordy a klasy, dziedziczenie, dynamiczny przydział pamięci, listy;

* Metody numeryczne: reprezentacja liczb rzeczywistych - standard IEEE 754-1985, stabilność numeryczna, metoda Newtona i pierwiastek kwadratowy, szeregi Maclaurina i funkcje trygonometryczne;

* Złożoność obliczeniowa: notacja duże O, rodzaje złożoności obliczeniowej, złożoność sortowania za pomocą sterty (ang. heap), metoda "na siłę" (ang. brute force) i jej złożoność, problemy optymalizacyjne i decyzyjne, transformacja wielomianowa problemów;

* Bazy danych: przetwarzanie plików, relacyjny model danych, język SQL, projekcja, selekcja, łączenie tabel, pielęgnacja danych, wzorce, zagnieżdżanie zapytań, grupowanie, manipulowanie bazą danych z poziomu języka Python;

* Uczenie maszynowe: główne podejścia do uczenia maszynowego, koncepcja uczenia nadzorowanego, entropia informacji, metoda ID3;

* Programowanie współbieżności: ewolucja systemów operacyjnych, procesy a wątki, podział czasu procesora, interferencja obliczeń, standard POSIX, biblioteka pthreads, zarządzanie wątkami, synchronizacja dostępu, semaforey, problem producent-konsument, problem czytelników i pisarzy, problem zakleszczenia;

* Sieci komputerowe: architektura warstwowa sieci komputerowej, komunikacja w sieci, stos protokołów internetowych, pakiety, protokół TCP/IP i gniazda TCP, protokół HTTP, język HTML i JavaScript, budowa serwisów webowych;

* Inżynieria oprogramowania: analiza problemu, wymagania funkcjonalne i przypadki użycia, wybrane diagramy języka UML, wstępny projekt interfejsu użytkownika, wymagania pozafunkcjonalne, architektura, implementacja i testowanie;

* Profesjonalizm w informatyce: zasady skutecznego działania, zasada win-win, synergia, licencje dot. oprogramowania, patenty.

Metody dydaktyczne

Wykład: prezentacja multimedialna i quizami

Ćwiczenia laboratoryjne: rozwiązywanie zadań, programowanie, dyskusja

Literatura

Podstawowa

1. Język ANSI C, B. W. Kernighan, D. M. Ritchie, Wydawnictwa Naukowo-Techniczne, dowolne wydanie
2. Układy cyfrowe, B. Wilkinson, WKiŁ, Warszawa, 2000
3. Wprowadzenie do przetwarzania tekstów w języku AWK, J. Nawrocki, W. Complak, ProDialog 2, 23-46, Poznań, 1994.
4. Sieci komputerowe, J.F. Kurose, K.W. Ross, Helion, 2006

Uzupełniająca

1. 7 nawyków skutecznego działania, S. Covey, Rebis, 2003
2. Język C. Szkoła programowania, Wydanie VI, Prata S., Helion, 2016

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	48	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	52	2,00