

--	--	--

<b>1. Nazwa przedmiotu: INFORMATYKA</b>		<b>2. Kod przedmiotu:</b>		
<b>3. Karta przedmiotu ważna od roku akademickiego: 2012</b>				
<b>4. Forma kształcenia:</b> studia pierwszego stopnia				
<b>5. Forma studiów:</b> studia stacjonarne				
<b>6. Kierunek studiów:</b> BIOTECHNOLOGIA; STUDIA MIĘDZYWYDZIAŁOWE				
<b>7. Profil studiów:</b> ogólnoakademicki				
<b>8. Specjalność:</b> BIOTECHNOLOGIA W OCHRONIE ŚRODOWISKA, BIOTECHNOLOGIA PRZEMYSŁOWA, BIOINFORMATYKA				
<b>9. Semestr:</b> 3, 4				
<b>10. Jednostka prowadząca przedmiot:</b> Instytut Automatyki, RAu1				
<b>11. Prowadzący przedmiot:</b> prof. dr hab. inż. Mieczysław Metzger				
<b>12. Przynależność do grupy przedmiotów:</b> przedmioty wspólne				
<b>13. Status przedmiotu:</b> obowiązkowy				
<b>14. Język prowadzenia zajęć:</b> polski				
<b>15. Przedmioty wprowadzające oraz wymagania wstępne:</b> matematyka; obsługa komputera, umiejętność korzystania z internetu, znajomość bioprocessów.				
<b>16. Cel przedmiotu:</b> Celem przedmiotu jest praktyczne nauczenie studentów programowania strukturalnego obliczeń inżynierskich na komputerach standardu PC, przy wykorzystaniu języka C/C++; języka m w środowisku Matlab oraz języka R w środowisku Bioconductor. Dodatkowym celem jest pokazanie najprostszych metod numerycznych w tym metod numerycznego rozwiązywania równań różniczkowych dla symulacji prostych bioprocessów.				
<b>17. Efekty kształcenia:</b>				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
W1	zna metodykę programowania strukturalnego na przykładzie języków C/C++, m oraz R. Zna też ogólne mechanizmy programowania charakterystyczne dla wszystkich języków programowania strukturalnego	CL, PS, OS	WM, L	K_W06, K_W21
W2	zna podstawowe metody programowania inżynierskich obliczeń komputerowych w tym podstaw programowania badań symulacyjnych	CL, PS, OS	WM, L	K_W02, K_W20, K_W21, K_W22
W3	Ma wiedzę o podstawowych sposobach programowania strukturalnego w językach C/C++; m oraz R.	CL, PS, OS	WM, L	K_W21
U1	Potrafi programować podstawowe obliczenia inżynierskie w językach C/C++; m oraz R.	CL, PS, OS	L	K_U06, K_U09, K_U26
U2	Potrafi wykorzystać znajomość podstawowych mechanizmów programowania strukturalnego do szybkiego opanowania dalszych języków programowania.	SP, CL	L	K_U04, K_U09, K_U26
K1	Potrafi samodzielnie podejmować decyzje dotyczące wyboru mechanizmów i algorytmów programowania	CL, PS	L	K_K07
K2	Potrafi zaprezentować i obronić wykonany samodzielnie program.	OS	L	K_K03

## 18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)

W. : 60 L.: 60

## 19. Treści kształcenia:

### Wykład

#### Semestr 3

**Wstęp do programowanie w językach wyższego rzędu:** Konieczność umiejętności programowania w pracy inżynierskiej; rola i znaczenie języków wyższego rzędu;

**Rola języka C we współczesnej informatyce:** C jako język łączący cechy języków wyższego i niższego rzędu; wiodąca rola języka C we współczesnym programowaniu; podkreślenie, że na bazie C uczymy się mechanizmów programowania wspólnych dla wszystkich języków programowania.

**Programowanie w języku C:** Idea programowania strukturalnego. Deklaracje globalne i lokalnie; Standardowe typy zmiennych; Wyrażenia; Wymuszenie typu wyrażenia; Instrukcje proste: instrukcje podstawienia; Komentarze; Instrukcje strukturalne: warunkowe; Instrukcje iteracyjne – pętle (for; while; do...while); Instrukcja switch; Omówienie wprowadzania i wyprowadzania danych; Biblioteki <stdio.h>, <iostream.h>; Omówienie standardu dla we/wy domyślnych: printf; scanf; formatowanie. Omówienie idei strumieni dla C++ Borlanda oraz DevC++; Łatwiejsza obsługa i lepsza kontrola we/wy; Strumienie cout, cin, cerr; Operatory << oraz >>; Wstawianie i wyjmowanie elementów ze strumienia; Więcej o operatorach i ich hierarchii; Zmiana typu wyrażenia; Uwaga na typ wyniku przy dzieleniu. Wprowadzenie do wskaźników i referencji. Wprowadzenie do funkcji; Ogólna postać programu z funkcjami; Cechy języka; Wykorzystanie funkcji w wyrażeniach; Wprowadzanie danych do funkcji (przez wartość); Wyprowadzanie wyniku – return; Parametry formalne i aktualne; Zakres działania; Deklaracja prototypów funkcji; Omówienie deklaracji static double; Zaslania; Przekazywanie danych przez globalność; Przekazywanie przez parametry aktualne; Program ilustrujący globalność i lokalność. Więcej o wykorzystaniu wskaźników; Komunikacja z funkcją przy użyciu wskaźników; wskaźniki i referencje; Programy przykładowe. Złożone struktury danych; tablice jedno i wielowymiarowe; Deklaracja typu tablicowego; Nazwa tablicy jako wskaźnik; Gromadzenie danych w tablicy; Wyciąganie danych z tablicy; Elementy tablic w wyrażeniach i funkcjach; Obsługa plików dyskowych; Deklaracja pliku FILE \*we; Funkcje fopen, fprintf, fscanf. Otwieranie plików; Zapis do pliku; Odczyt z pliku; Plik tekstowy jako pośrednictwo z innymi programami; otwarcie pliku ze sprawdzeniem obecności; Program Przebieg; Operacje na plikach w c++ – biblioteka <fstream.h>; Kreacja i otwarcie pliku; sterowanie strumieniami; Zamykanie pliku. Zmienne dynamiczne, dynamiczne struktury danych.

**Wstęp do metod numerycznych:** ogólne zasady obliczeń komputerowych; rozwiązanie przybliżone i definicje błędów obliczeniowych; wybrane metody numeryczne dla obliczeń inżynierskich; metody iteracyjne i metody rekurencyjne; numeryczne rozwiązywanie równań nieliniowych; numeryczne całkowanie równań różniczkowych; generacja wymuszeń (sinusoidalnych, prostokątnych i pseudolosowych);

**Podstawy symulacji procesów biotechnologicznych:** standard UDYN; prosty element inercyjny; właściwości uśredniające elementu inercyjnego dla periodycznych zmian sygnału wejściowego.

#### Semestr 4.

**Podstawowe programy użytkowe:** Rola programów użytkowych w pracach inżynierskich; programowanie obliczeń inżynierskich w środowisku Matlab;

**Programowanie w środowisku Matlab:** Ogólna ideologia oprogramowania Matlab. Pokaz inicjacji tablic i najprostszyc obliczeń. Powiększanie macierzy; Wycinanie kawałków; Operacje macierzowe i tablicowe; Funkcje i ich wykorzystanie; Globalność w środowisku Matlab; Wprowadzanie danych do funkcji; Zwracanie wyników; Grafika w Matlabie; Pokaz bardziej skomplikowanych programów: iteracyjne rozwiązanie nieliniowego równania algebraicznego. Wykorzystanie Matlab do wykresów z programu obliczającego przebieg X(t) w C (zamiast przebiegu z doświadczenia) – pośrednictwo jawnych plików tekstowych (można łatwo coś zmienić). Wklejenie wykresów do Worda. Atrybuty obiektów graficznych; Demo zaawansowanych programów (mini-środowisk windowsowych). Prezentacja wybranych toolboxów.

**Symulacja bioprocessów:** realizacja standardu UDYN w Matlabie; zbiorniki z wymuszonymi przepływami; Zbiornik wypełniony z przelewem; zbiorniki z samoregulacją; właściwości uśredniające zbiornika stabilizującego dla periodycznych zmian stężeń wejściowych; Model 2 zbiorników wypełnionych (często występuje w oczyszczalniach); bilanse substancji i energii; Mieszalniki; Bioreaktory; Prosty bioreaktor przepływowy dla redukcji substratu; Chemostat; konkurencja kultur w chemostacie; możliwości koegzystencji kultur w chemostacie; Problem obliczenia warunku początkowego i ustalonego punktu pracy.

**Bioconductor i język R jako podstawowe narzędzie biotechnologa:** podstawy ideowe; instalacja środowiska obliczeniowego; uruchomienie podstawowych elementów.

**Programowanie w języku R:** Ideologia Bioconductora i środowisko języka R, wprowadzenie, wektory i operacje na nich, programy, bloki i instrukcje sterujące; Najprostsze wektory można utworzyć intuicyjnie - funkcja powiązania c(); Utrzymywanie danych w przestrzeni roboczej i ich eliminacja; Operacje arytmetyczne na wektorach; Funkcje operujące na wektorach; Struktury i instrukcje sterujące; Bloki: {wyrażenie\_1, ..., wyrażenie\_n} – blok jest też wyrażeniem a więc może być użyte w instrukcjach złożonych; instr warunkowa: if (wyr\_logiczne) wyr\_2 else wyr\_3 (używamy jak w C/C++ operatorów && oraz ||); pętla for: for (nazwa\_zmien\_petli in wyr\_1) wyr\_2 (wyr\_1 to często np. 1:100; wyr\_2 to typowo blok); pętla powtarzaj: repeat wyrażenie (aby zakonczyc w wyrażeniu musi się pojawić break); pętla while: while (warunek) wyrażenie; Pisanie i wykonywanie programów; Obliczenia statystyczne, odczyt i zapis danych na dysk.

## Ćwiczenia tablicowe

nie ma

## Zajęcia laboratoryjne

### Semestr 3

1. Wprowadzenie do laboratorium (zapoznanie ze środowiskiem programowania)
2. Podstawy języka C (wprowadzanie danych, zapis wyrażeń algebraicznych)
3. Wykorzystanie instrukcji warunkowych (instrukcje if oraz switch)
4. Instrukcje iteracyjne cz. 1 (instrukcje while, for i do...while)
5. Instrukcje iteracyjne cz. 2 (instrukcje break i continue)
6. Instrukcje iteracyjne cz. 3 (kontynuacja tematu)
7. Złożone struktury danych cz. 1 (tablice jednowymiarowe)
9. Złożone struktury danych cz. 2 (tablice wielowymiarowe)
10. Złożone struktury danych cz. 3 (zapis i odczyt danych z plików)
11. Funkcje cz. 1 (sposób tworzenia funkcji i ich wywoływania, przekazywanie parametrów przez wartość)
12. Funkcje cz. 2 (przekazywanie parametrów poprzez wskaźnik i referencję)
13. Złożone konstrukcje programowe - rekurencja
14. Kolokwium sprawdzające cz. 1
15. Kolokwium sprawdzające cz. 2

### Semestr 4

1. MATLAB – podstawy
2. MATLAB – tworzenie funkcji
3. MATLAB – wielomiany, interpolacja
4. MATLAB – regresja, statystyka
5. MATLAB – obliczenia symboliczne
6. MATLAB – symulacja bioprocessów – zbiorniki i ich połączenia
7. MATLAB – symulacja bioprocessów - bioreaktor
8. R – Proste obliczenia w środowisku języka R.
9. R – Wektory i operacje na nich, obliczenia i wykresy przebiegów, programy i ich egzekucja, struktury warunkowe i iteracyjne.
10. R – Złożone struktury danych w języku R
11. R – Funkcje w języku R
12. R – statystyka i grafika, zapis i odczyt z pliku
13. R – symulacja bioprocessów
14. Kolokwium sprawdzające cz. 1
15. Kolokwium sprawdzające cz. 2

**20. Egzamin:** tylko zaliczenie

## 21. Literatura podstawowa:

1. B.Kernigham, D.Ritchie: Język C (7 wydań)
2. J. Grębosz: *Symfonia C++*. Programowanie w języku C++ orientowane obiektowo. Oficyna Kallimach, Kraków1999
3. Matlab user-manual
4. Język R - Bioconductor user manual

## 22. Literatura uzupełniająca:

1. B. Stroustrup: *Język C++*. WNT Warszawa 1994
2. B. Stroustrup: *Projektowanie i rozwój języka C++*. WNT Warszawa,1994

**23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia:**

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	60/30 (60)
2	Ćwiczenia	0/0
3	Laboratorium	60/30 (100)
4	Projekt	0/0
5	Seminarium	0/0
6	Inne (egzamin i przygotowanie)	15/15 (20/40)
	Suma godzin	135/75 (200)
<b>24. Suma wszystkich godzin: 210 (340)</b>		
<b>25. Liczba punktów ECTS: 7 (8)</b>		
<b>26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 4</b>		
<b>27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 3</b>		
<b>26. Uwagi:</b>		

Zatwierdzono:

.....  
(data i podpis prowadzącego)

.....  
(data i podpis dyrektora instytutu/kierownika katedry/  
Dyrektora Kolegium Języków Obcych/kierownika lub  
dyrektora jednostki międzywydziałowej)