

SYLABUS PRZEDMIOTU: Programowanie I i II

L.p.	Elementy składowe sylabusu	Opis
1.	Nazwa przedmiotu	Programowanie I i II
2.	Nazwa jednostki prowadzącej przedmiot	Wydział Matematyki i Informatyki, Instytut Matematyki
3.	Kod przedmiotu	
4.	Język przedmiotu	Język polski
5.	Grupa treści kształcenia, w ramach której przedmiot jest realizowany	Przedmiot realizowany w ramach grupy treści kierunkowych.
6.	Typ przedmiotu	Przedmiot obowiązkowy do ukończenia całego toku studiów.
7.	Rok studiów, semestr	Rok I, semestr I i II, specjalność komputerowa
8.	Imię i nazwisko osoby (osób) prowadzącej przedmiot	
9.	Imię i nazwisko osoby (osób) egzaminującej bądź udzielającej zaliczenia w przypadku, gdy nie jest nim osoba prowadząca dany przedmiot	
10.	Formuła przedmiotu	Wykład i ćwiczenia
11.	Wymagania wstępne	Brak
12.	Liczba godzin zajęć dydaktycznych	30 godzin wykładu i 30 godzin ćwiczeń w semestrze I oraz 30 godzin wykładu i 45 godzin ćwiczeń w semestrze II
13.	Liczba punktów ECTS przypisana przedmiotowi	11
14.	Czy podstawa obliczenia średniej ważonej?	Przedmiot stanowi podstawę obliczenia średniej ważonej.

15.	Założenia i cele przedmiotu	Celem kursu jest opanowanie przez studentów podstaw klasycznych i współczesnych technik programowania, w tym programowania strukturalnego, obiektowego, obiektowo orientowanego i funkcjonalnego na bazie wybranych języków programowania ze szczególnym uwzględnieniem języków C, C++, C#, Java, Scheme i PERL. Ukończenie kursu daje studentom solidne podstawy do uczestnictwa we wszystkich innych kursach przewidzianych programem studiów, wymagających umiejętności programowania.
16.	Metody dydaktyczne	Wykład prowadzony jest w formie prezentacji wspieranej licznymi przykładami programów. Ćwiczenia prowadzone są w pracowni komputerowej. Studenci rozwiązują zadania programistyczne weryfikowane przez automatyczny system sprawdzający oraz przez asystentów. Studenci przygotowują też swój pierwszy większy projekt programistyczny oceniany przez asystentów.
17.	Forma i warunki zaliczenia przedmiotu, w tym zasady dopuszczenia do egzaminu, zaliczenia z przedmiotu, a także formę i warunki zaliczenia poszczególnych form zajęć wchodzących w zakres danego przedmiotu	Przedmiot w semestrze I kończy się zaliczeniem ćwiczeń na ocenę, zaś w semestrze drugim kończy się egzaminem pisemnym i/lub ustnym. Do podejścia do egzaminu konieczne jest zaliczenie ćwiczeń. Podstawą uzyskania zaliczenia z ćwiczeń jest ocenianie ciągle i/lub kilka (ilość zależy od prowadzących ćwiczenia) pisemnych sprawdzianów.
18.	Treści merytoryczne przedmiotu oraz sposób ich realizacji	Wprowadzenie: historia rozwoju języków i technik programowania, języki wysokiego poziomu, translacja, kompilatory i interpretry, przegląd podstawowych koncepcji języków programowania, elementy programowania strukturalnego w języku C i językach pochodnych: C++, Java, C#. Programowanie niskopoziomowe: obiekty instrukcje i podprogramy na poziomie procesora, assembler, obsługa przerwań i stosu, kompilacja modułowa i linkowanie. Programowanie proceduralne: wyrażenia, notacje funkcyjne, ewaluacja wyrażenia, programowanie poprzez stos, podprogramy, funkcje i procedury. Programowanie strukturalne: zasady programowania strukturalnego, instrukcje selekcji, instrukcje pętli. Podstawy programowania funkcjonalnego: rekursja, zasady programowania funkcjonalnego, wyrażenia listowe i notacja listowa, funkcje anonimowe. Typy danych: typy podstawowe i definiowane, typy pochodne i typy złożone, kontrola typów, polimorfizm, typy wartościowe i referencyjne. Operatory: operatory matematyczne, operatory konwersji, kolejność wykonywania operatorów i porządek wartościowania. Zmienne i tablice: zmienna jako nazwany obiekt, deklarowanie zmiennej, zakres ważności nazwy, inicjalizacja zmiennych, czas życia obiektu, obiekty stałe, typ tablicowy, tablice wielowymiarowe i tablice tablic, tablice nieregularne. Odnośniki: referencje, wskaźniki, arytmetyka wskaźników, typy referencyjne, zastosowania, sarta. Funkcje: przekazywanie argumentów, obiekty chwilowe, zwracanie wartości, przeładowanie nazw funkcji. Zmienne, funkcje i odnośniki w kontekście kompilacji. Programowanie bazujące na obiektach: złożoność strukturalna, abstrakcja danych, klasa i obiekt, enkapsulacja, projekt, interfejs i implementacja. Konstruktory i destruktory, konstruktory kopiujące, jawne i niejawne wywołanie konstruktora. Zarządzanie nazwami: klasy zagnieżdżone i klasy wewnętrzne, klasy lokalne, przestrzenie nazw i pakiety. Przeładowanie operatorów: operator jako funkcja, przeładowanie operatorów jako funkcji globalnych oraz metod, zastosowania przeładowania operatorów Konwersje: konwersje standardowe, konwertery i funkcje konwertujące, konwersje jawne i niejawne, niejednoznaczność konwersji. Podstawy programowania obiektowo orientowanego: dziedziczenie, dziedziczenie wielopokoleniowe, hierarchia klas, dostęp do składników w kontekście dziedziczenia, konstrukcja obiektów w kontekście dziedziczenia, dziedziczenie

		wielokrotne. Funkcje wirtualne: mechanizm wirtualności, korzyści i koszty wirtualności, wczesne i późne wiązanie, wirtualna konstrukcja i destrukcja obiektów, polimorfizm dynamiczny w kontekście funkcji wirtualnych. Klasy abstrakcyjne: metody abstrakcyjne, cechy klasy abstrakcyjnej, korzyści z klasy abstrakcyjnej, interfejsy, siła klas abstrakcyjnych, istota programowania obiektowo orientowanego. Identyfikacja typów w trakcie wykonania (RTTI): bezwzględne i relatywne RTTI, niebezpieczeństwa związane z RTTI, zastosowania RTTI, wielometody. Obsługa sytuacji wyjątkowych: sytuacje wyjątkowe, rzucanie wyjątków, łapanie wyjątków, informowanie o rzucanych wyjątkach, hierarchie klas do przechowywania informacji o wyjątkach, sprzątanie stosu, pozyskiwanie zasobów poprzez inicjalizację. Wprowadzenie do programowania generycznego: szablony funkcji i klas, klasy i metody generyczne, polimorfizm statyczny. Pojemniki: pojemniki sekwencyjne i asocjacyjne, typy pojemników, iteratory. Programowanie funkcyjne: funkcjonały, currying, klasy i obiekty funkcyjne, zalety i wady programowania funkcjonalnego. Programowanie orientowane zdarzeniami: pętla główna, zdarzenia, obsługa zdarzeń. Programowanie wielowątkowe: wątki, współdzielenie zasobów, synchronizacja, komunikacja pomiędzy wątkami, klincz. Graficzny interfejs użytkownika: komponenty i kontenery, okna dialogowe, menedżery układu okien, systemy menu, komponenty tekstowe i graficzne. Operacje wejścia-wyjścia: strumienie, operacje na strumieniach, manipulatory, formatowane i nieformatowane operacja wejścia-wyjścia, strumienie plikowe. Języki skryptowe. Wyrażenia regularne: wzorce, operator dopasowania, operator podmiany, podwzorce. Języki do przechowywania i przetwarzania dokumentów: znaczniki, atrybuty, poprawność składni, formatowanie, formularze Programowanie w środowisku przeglądarki: języki skryptowe w środowisku przeglądarki, Document Object Model, zdarzenia, aplety.
19.	Wykaz literatury podstawowej i uzupełniającej, obowiązującej do zaliczenia danego przedmiotu	T. Christiansen, N. Torkington, Perl receptury, Wydawnictwo RM, Warszawa 1998 P. Coad, J. Nicola, Object-oriented Programming, Yourdon Press, Prentice Hall, New Jersey, 1993 D. Conway, Object Oriented Perl, Manning Publications Co., Greenwich, Connecticut, 2000. B. Eckel, Thinking in C++ T 1 i 2, Helion, Gliwice, 2002-2004 B. Eckel, Thinking in Java, Helion, Gliwice, 2003 J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków, 1996 N.M. Josuttis, C++ Biblioteka Standardowa, Helion, Gliwice, 2003. A. Hejlsberg, S. Wiltamuth, P. Golde, The C# Programming Language, Addison-Wesley, Boston, 2006. B. W. Kernighan, D.M. Ritchie, Język C, WNT, Warszawa 1987 J. Liberty, Programowanie C#, Helion, Gliwice, 2006. B.J. MacLean, Functional Programming, Addison Wesley, New York, 1990. P. Naughton, H. Schildt, The Complete Reference Java, Osborne, McGraw-Hill, Berkeley, California, 1997 M.L. Scott, Programming Language Pragmatics, Elsevier, Boston, 2006. R.W. Sebesta, Concepts of Programming Languages, Addison Wesley, Boston, 2004 B. Stroustrup, Język C++, WNT, Warszawa 2000 D. Vandevoorde, N.M. Josuttis, C++ Szablony, Helion, Gliwice, 2003