

Nazwa modułu kształcenia	Programowanie 1		
Nazwa jednostki prowadzącej moduł	Instytut Informatyki, Wydział Matematyki i Informatyki		
Kod modułu	WMI.II. P1-OL		
Język kształcenia	Polski		
Efekty kształcenia dla modułu kształcenia	Symbol	Efekty kształcenia	Odniesienie do efektów kierunkowych
	E1	zna podstawowe techniki programowania obiektowego oraz projektowania programów z użyciem klas i obiektów	K_W06+++
	E2	zna podstawowe koncepcje programowania funkcyjnego	K_W06++
	E3	potrafi korzystać z debuggera oraz zintegrowanych środowisk programistycznych	K_W05+, K_U09+
	E4	potrafi poprawnie zaprojektować oprogramowanie zgodnie z metodyką obiektową	K_U10++
	E5	potrafi zaprojektować i zaimplementować prosty model informatyczny na podstawie podanej specyfikacji	K_U16+++
	E6	potrafi ustnie i pisemnie przedstawić rozwiązanie prostego problemu informatycznego	K_U20++, K_U21+
	E7	rozumie i docenia znaczenie wartości intelektualnej w działaniach własnych oraz innych osób	K_K06++
Typ modułu kształcenia (obowiązkowy/fakultatywny)	Obowiązkowy dla następujących specjalności studiów I stopnia: 1. informatyka stosowana 2. inżynieria oprogramowania 3. modelowanie, sztuczna inteligencja i sterowanie 4. matematyka komputerowa		
Rok studiów	I		
Semestr	1		
Imię i nazwisko osoby/osób prowadzących moduł	Prof. dr hab. Marian Mrozek, dr hab. Daniel Wilczak, dr Paweł Dłotko, dr Paweł Gniadek, dr Tomasz Kapela, dr Marcin Żelawski		

Imię i nazwisko osoby/osób egzaminującej/egzaminujących bądź udzielającej zaliczenia, w przypadku gdy nie jest to osoba prowadząca dany moduł	Prof. dr hab. Marian Mrozek, dr hab. Daniel Wilczak
Sposób realizacji	wykład, laboratorium
Wymagania wstępne i dodatkowe	Wstęp do Programowania
Liczba godzin zajęć dydaktycznych wymagających bezpośredniego udziału nauczyciela akademickiego i studentów, gdy w danym module przewidziane są takie zajęcia	60
Liczba punktów ECTS przypisana modułowi	5
Bilans punktów ECTS	Udział w wykładach - 30 godz. Udział w zajęciach laboratoryjnych – 30 godz. Samodzielna implementacja zadań programistycznych – 75 godz. Łączny nakład pracy studenta: 135 godzin , co odpowiada 5 punktom ECTS
Stosowane metody dydaktyczne	1. Wykład ilustrowany prezentacją komputerową. 2. Ćwiczenia w laboratorium komputerowym, połączone z dyskusją przy tablicy. 3. Samodzielna implementacja zadań programistycznych i automatyczne testowanie ich na serwerze zadaniowym
Metody sprawdzania i oceny efektów kształcenia uzyskanych przez studentów	Programistyczne zadania domowe (E1, E2 E3,E4, E5) Automatycznie weryfikowane programy zaliczeniowe (E1, E2, E3, E4, E5) Prezentowanie rozwiązań zadań domowych oraz ustna obrona rozwiązań zadań programistycznych wysłanych do serwera automatycznej weryfikacji (E6, E7) Wykorzystanie systemu antyplagiatowego na serwerze weryfikującym zadania programistyczne (E7)

Forma i warunki zaliczenia modułu, w tym zasady dopuszczenia do egzaminu, zaliczenia, a także forma i warunki zaliczenia poszczególnych zajęć wchodzących w zakres danego modułu	<p>Student otrzymuje ocenę końcową z ćwiczeń na podstawie punktów przyznawanych za systematycznie oddawane zadania domowe oraz programy oceniane automatycznie przez system weryfikacji programów. Warunkiem otrzymania zaliczenia ćwiczeń jest uzyskanie łącznie co najmniej 50% możliwych do zdobycia punktów.</p>
Treści modułu kształcenia	<ol style="list-style-type: none">1. Wprowadzenie: historia rozwoju języków i technik programowania, języki wysokiego poziomu, translacja, kompilatory i interpretery, przegląd podstawowych koncepcji języków programowania2. Elementarne wprowadzenie w programowanie w języku C++3. Programowanie niskopoziomowe: obiekty instrukcje i podprogramy na poziomie procesora, asembler, obsługa przerwań i stosu, kompilacja modułowa i linkowanie4. Programowanie proceduralne: wyrażenia, notacje funkcyjne, ewaluacja wyrażenia, programowanie poprzez stos, podprogramy, funkcje i procedury5. Programowanie strukturalne: zasady programowania strukturalnego, instrukcje selekcji, instrukcje pętli6. Podstawy programowania funkcjonalnego: rekursja, zasady programowania funkcjonalnego, wyrażenia listowe i notacja listowa, funkcje anonimowe7. Typy danych: typy podstawowe i definiowane, typy pochodne i typy złożone, kontrola typów, polimorfizm, typy wartościowe i referencyjne8. Operatory: operatory matematyczne, operatory konwersji, kolejność wykonywania operatorów i porządek wartościowania9. 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 nieregularne10. Odnośniki: referencje, wskaźniki, arytmetyka wskaźników, typy referencyjne, zastosowania, sarta11. Funkcje: przekazywanie argumentów, obiekty chwilowe, zwracanie wartości, przeładowanie nazw funkcji12. Programowanie bazujące na obiektach: złożoność strukturalna, abstrakcja danych, klasa i obiekt, enkapsulacja, projekt, interfejs i implementacja

	<p>13. Konstruktory i destruktory, konstruktory kopiujące, jawne i niejawne wywołanie konstruktora</p> <p>14. Zarządzanie nazwami: klasy zagnieżdżone i klasy wewnętrzne, klasy lokalne, przestrzenie nazw i pakiety</p> <p>15. Przeładowanie operatorów: operator jako funkcja, przeładowanie operatorów jako funkcji globalnych oraz metod, zastosowania przeładowania operatorów</p> <p>16. Konwersje: konwersje standardowe, konwertery i funkcje konwertujące, konwersje jawne i niejawne, niejednoznaczność konwersji</p>
<p>Wykaz literatury podstawowej i uzupełniającej, obowiązującej do zaliczenia danego modułu</p>	<p>Moduł ma charakter autorski, obowiązuje przede wszystkim materiał wyłożony, literatura ma charakter pomocniczy.</p> <ol style="list-style-type: none">1. B. Eckel, Thinking in C++ T 1 i 2, Helion, Gliwice, 2002-20042. B. Eckel, Thinking in Java, Helion, Gliwice, 20033. J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków, 19964. N.M. Josuttis, C++ Biblioteka Standardowa, Helion, Gliwice, 2003.5. A. Hejlsberg, S. Wiltamuth, P. Golde, The C# Programming Language, Addison-Wesley, Boston, 2006.6. W. Kernighan, D.M. Ritchie, Język C, WNT, Warszawa 19877. J. Liberty, Programowanie C#, Helion, Gliwice, 2006.8. B.J. MacLean, Functional Programming, Addison Wesley, New York, 1990.9. P.Naughton, H. Schildt, The Complete Reference Java, Osborne, McGraw-Hill, Berkeley, California, 199710. M.L. Scott, Programming Language Pragmatics, Elsevier, Boston, 2006.11. R.W. Sebesta, Concepts of Programming Languages, Addison Wesley, Boston, 200412. B. Stroustrup, Język C++, WNT, Warszawa 2000
<p>Metody i kryteria oceniania</p>	<p>Metody oceniania:</p> <ol style="list-style-type: none">1. zadania automatycznie weryfikowane2. prezentacja zadań domowych <p>Kryteria oceniania:</p> <ol style="list-style-type: none">1. za każde zadanie automatycznie weryfikowane można zdobyć określoną maksymalną liczbę punktów. Ilość zdobytych punktów zależy od zaliczonych przez rozwiązanie testów.2. zadania domowe są oceniane przez prowadzącego ćwiczenia po ustnej prezentacji rozwiązania. <p>Końcowa ocena z ćwiczeń jest średnią ważoną wyników uzyskanych za poszczególne elementy. Proporcje ważenia są ustalane przez wykładowcę.</p>

Wymiar, zasady i forma odbywania praktyk, w przypadku, gdy program kształcenia przewiduje praktyki	Nie dotyczy
---	-------------