<table>
<tr><td>Nazwa modułu kształcenia</td><td colspan="3">Programowanie 2</td></tr>
<tr><td>Nazwa jednostki prowadzącej moduł</td><td colspan="3">Instytut Informatyki,
Wydział Matematyki i Informatyki</td></tr>
<tr><td>Kod modułu</td><td colspan="3">WMI.II- P2-OL</td></tr>
<tr><td>Język kształcenia</td><td colspan="3">Polski</td></tr>
<tr><td>Efekty kształcenia dla modułu kształcenia</td><td colspan="3"><table>
<tr><td>Symbol</td><td>Efekty kształcenia</td><td>Odniesienie do efektów kierunkowych</td></tr>
<tr><td>E1</td><td>zna zaawansowane techniki programowania obiektowego i obiektowo orientowanego oraz generycznego</td><td>K_W06+++</td></tr>
<tr><td>E2</td><td>potrafi korzystać z debuggera, zintegrowanych środowisk programistycznych oraz dzielić program na moduły</td><td>K_W05++, K_U09++</td></tr>
<tr><td>E3</td><td>potrafi poprawnie zaprojektować oprogramowanie zgodnie z metodyką obiektową oraz z wykorzystaniem typów generycznych</td><td>K_U10+++, K_U14+</td></tr>
<tr><td>E4</td><td>zna biegle C++ oraz w stopniu zaawansowanym Java oraz C#</td><td>K_W06++, K_U04++</td></tr>
<tr><td>E5</td><td>potrafi zaprojektować złożony model informatyczny na podstawie podanej specyfikacji</td><td>K_U16+</td></tr>
<tr><td>E6</td><td>potrafi ustnie i pisemnie przedstawić rozwiązanie złożonego problemu informatycznego</td><td>K_U20+, K_U21+</td></tr>
<tr><td>E7</td><td>rozumie i docenia znaczenie wartości intelektualnej w działaniach własnych oraz innych osób</td><td>K_K06++</td></tr>
</table></td></tr>
<tr><td>Typ modułu kształcenia (obowiązkowy/fakultatywny)</td><td colspan="3">Obowiązkowy dla następujących specjalności studiów I stopnia:
1. informatyka stosowana
2. inżynieria oprogramowania
3. modelowanie, sztuczna inteligencja i sterowanie
4. matematyka komputerowa</td></tr>
<tr><td>Rok studiów</td><td colspan="3">I</td></tr>
<tr><td>Semestr</td><td colspan="3">2</td></tr>
<tr><td>Imię i nazwisko osoby/osób</td><td colspan="3">Prof. dr hab. Marian Mrozek, dr hab. Daniel Wilczak, dr Paweł Dłotko, dr Paweł Gniadek, dr Tomasz Kapela,</td></tr>
</table>

prowadzących moduł	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, Programowanie 1
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	**8**
Bilans punktów ECTS	Udział w wykładach - 30 godz. Udział w zajęciach laboratoryjnych – 30 godz. Samodzielna implementacja zadań programistycznych – 120 godz. Przygotowanie do egzaminu, analiza programów do egzaminu oraz udział w egzaminie – 30 godz. Łączny nakład pracy studenta: 210 godzin , co odpowiada 8 punktom ECTS
Stosowane metody dydaktyczne	1. Wykład ilustrowany prezentacją komputerową. 2. Ćwiczenia w laboratorium komputerowym, połączone z dyskusją przy tablicy. 3. Samodzielne 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) Egzamin (E1, E4, E6, E7) 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	Student otrzymuje ocenę końcowa 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. Każdy student jest dopuszczany do egzaminu bez względu na wynik uzyskany na ćwiczeniach. Ocena końcowa z kursu jest wyznaczana na podstawie średniej ważonej wyników procentowych z ćwiczeń oraz egzaminu. Wagi ustala wykładowca na początku kursu.
Treści modułu kształcenia	1. Podstawy programowania obiektowo orientowanego: dziedziczenie, dziedziczenie wielopokoleniowe, hierarchia klas, dostęp do składników w kontekście dziedziczenia, konstrukcja obiektów w kontekście dziedziczenia, dziedziczenie wielokrotne 2. Funkcje wirtualne: mechanizm wirtualności, pożytki i koszty wirtualności, wczesne i późne wiązanie, wirtualna konstrukcja i destrukcja obiektów, polimorfizm dynamiczny w kontekście funkcji wirtualnych 3. Klasy abstrakcyjne: metody abstrakcyjne, cechy klasy abstrakcyjnej, pożytki z klasy abstrakcyjnej, interfejsy, siła klas abstrakcyjnych, istota programowania obiektowo orientowanego 4. Identyfikacja typów w trakcie wykonania (RTTI): bezwzględne i relatywne RTTI, niebezpieczeństwa związane z RTTI, zastosowania RTTI, wielometody. 5. 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 6. Wprowadzenie do programowania generycznego: szablony funkcji i klas, klasy i metody generyczne, polimorfizm statyczny 7. Pojemniki: pojemniki sekwencyjne i asocjacyjne, typy pojemników, iteratory. 8. Programowanie funkcjonalne: funkcjonały, currying, klasy i obiekty funkcyjne, zalety i wady programowania funkcjonalnego 9. Programowanie orientowane zdarzeniami: pętla główna, zdarzenia, obsługa zdarzeń 10. Programowanie wielowątkowe: wątki, współdzielenie zasobów, synchronizacja, komunikacja pomiędzy wątkami, klincz.

	11. Graficzny interfejs użytkownika: komponenty i kontenery, okna dialogowe, menedżery układu okien, systemy menu, komponenty tekstowe i graficzne 12. Operacje wejścia-wyjścia: strumienie, operacje na strumieniach, manipulatory, formatowane i nieformatowane operacja wejścia-wyjścia, strumienie plikowe
Wykaz literatury podstawowej i uzupełniającej, obowiązującej do zaliczenia danego modułu	Moduł ma charakter autorski, obowiązuje przede wszystkim materiał wyłożony, literatura ma charakter pomocniczy. 1. Christiansen, N. Torkington, Perl receptury, Wydawnictwo RM, Warszawa 1998 2. P. Coad, J. Nicola, Object-oriented Programming, Yourdon Press, Prentice Hall, New Jersey, 1993 3 D. Conway, Object Oriented Perl, Manning Publications Co., Greenwich, Connecticut, 2000. 4. B. Eckel, Thinking in C++ T 1 i 2, Helion, Gliwice, 2002-2004 5. B. Eckel, Thinking in Java, Helion, Gliwice, 2003 6. J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków, 1996 7. N.M. Josuttis, C++ Biblioteka Standardowa, Helion, Gliwice, 2003. 8. A. Hejlsberg, S. Wiltamuth, P. Golde, The C# Programming Language, Addison-Wesley, Boston, 2006. 9. W. Kernighan, D.M. Ritchie, Język C, WNT, Warszawa 1987 10. J. Liberty, Programowanie C#, Helion, Gliwice, 2006. 11. B.J. MacLean, Functional Programming, Addison Wesley, New York, 1990. 12. P.Naughton, H. Schildt, The Complete Reference Java, Osborne, McGraw-Hill, Berkeley, California, 1997 13. M.L. Scott, Programming Language Pragmatics, Elsevier, Boston, 2006. 14. R.W. Sebesta, Concepts of Programming Languages, Addison Wesley, Boston, 2004 15. B. Stroustrup, Język C++, WNT, Warszawa 2000 16. D. Vandevoorde, N.M. Josuttis, C++ Szablony, Helion, Gliwice, 2003
Metody i kryteria oceniania	Metody oceniania: 1. zadania automatycznie weryfikowane 2. prezentacja zadań domowych 3. egzamin w formie testu wielokrotnego wyboru Kryteria oceniania: 1. za każde automatycznie weryfikowane zadanie można zdobyć określoną maksymalną liczbę punktów. Ilość zdobytych punktów zależy od ilości zaliczonych przez rozwiązanie testów. 2. zadania domowe są oceniane przez prowadzącego ćwiczenia po ustnej prezentacji rozwiązania.

	Końcowa ocena kursu jest średnią ważoną wyników uzyskanych za poszczególne elementy. Wagi są ustalane przez wykładowcę na początku kursu.
Wymiar, zasady i forma odbywania praktyk, w przypadku, gdy program kształcenia przewiduje praktyki	Nie dotyczy