

PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA W NOWYM SĄCZU

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2014/2015

Instytut Techniczny

Kierunek studiów: Mechatronika

Profil: Ogólnoakademicki

Forma studiów: Stacjonarne

Kod kierunku: 06.0

Stopień studiów: I

Specjalności: Mechatronika stosowana

1 PRZEDMIOT

NAZWA PRZEDMIOTU	Języki programowania JAVA, C, C++
KOD PRZEDMIOTU	IT 06.0 AIS MP6 14/15
KATEGORIA PRZEDMIOTU	Przedmioty specjalnościowe
LICZBA PUNKTÓW ECTS	5
SEMESTRY	5

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	PROJEKT	SEMINARIUM
5	15			45	

3 CELE PRZEDMIOTU

Cel 1 Zapoznanie się z głównymi cechami trzech fundamentalnych języków programowania (C, C++, Java) oraz wspieranych przez nie paradygmatów programowania (proceduralnego, obiektowego, uogólnionego). Przegląd kompilatorów oraz metod kompilowania.

Cel 2 Opanowanie podstawowych składni językowych (typy i zmienne, operatory, operacje warunkowe, pętle), funkcje. Typy złożone (tablice, wskaźniki, referencje, struktury, klasy - oraz ich metody). Operacje wejścia / wyjścia (strumienie, pliki).

Cel 3 Operowanie technikami obiektowymi - mechanizmy dziedziczenia i polimorfizmu. Abstrakcja danych. Kontenery i kolekcje. Iteratory oraz podstawowe algorytmy uogólnione.

Cel 4 Poznanie operacji niskiego poziomu (obliczenia na bitach), bibliotek matematycznych oraz generatorów losowych, wspierających zagadnienia spotykane w mechatronice.



4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

- a Podstawowa umiejętność obsługi komputera (na poziomie użytkownika) w systemie Windows oraz w systemie Linux.

5 EFEKTY KSZTAŁCENIA

- EK1** Wiedza: Student potrafi opisać podstawowe składnie języków (C, C++, Java), objaśniać różnice pomiędzy tymi językami. Wskazuje na podobieństwa leksykalne oraz charakteryzuje typy, operatory, wyrażenia warunkowe i pętle. Potrafi zaproponować odpowiednie narzędzie (kompilator) i napisać, skompilować i uruchomić przykładowy program rozwiązujący proste zagadnienia numeryczne.
- EK2** Umiejętności: Student potrafi korzystać ze złożonych struktur danych, dostępnych w danym języku: tablic, wskaźników, referencji. Potrafi pisać struktury oraz klasy, reprezentujące obiekt, oraz funkcje i metody składowe.
- EK3** Umiejętności: Student dobiera paradygmaty programowania w zależności od rodzaju problemu. Potrafi planować obiektowe zależności i wykorzystywać mechanizmy polimorfizmu. Łączy metodologie skutecznie rozwiązujące konkretne zadania, korzysta z kontenerów czy kolekcji, wie jak używać iteratory. Zna podstawowe algorytmy z bibliotek, które podnoszą efektywność działania na danych.
- EK4** Umiejętności: Student efektywnie posługuje się operacjami bitowymi (w różnych systemach liczbowych). Potrafi wyszukać rozwiązania wyższego poziomu dostarczane w bibliotekach konkretnych języków (zwłaszcza matematycznych). Potrafi wykonać symulacje Monte-Carlo prostych zagadnień, używając generatorów losowych o różnych rozkładach,

6 TREŚCI PROGRAMOWE

WYKŁAD

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Omówienie podstawowych cech języków C, C++ i Java oraz paradygmatów programowania proceduralnego, obiektowego i uogólnionego, wspieranych przez te języki. Standardy oraz specyfikacje języków. Kompilatory i środowiska, w których można pisać, kompilować i wykonywać programy. Przenośność kodu.	2
W2	Podstawowe zasady programowania: co to jest program. Proste typy obsługiwane przez języki, zakres typów i ich modyfikatory. Pojęcie obiektów globalnych, lokalnych, statycznych. Modyfikator const. Operatory i ich hierarchie - matematyczne, bitowe, logiczne, rzutowania. Proste operacje wejścia i wyjścia.	2
W3	Typy złożone, porównanie pomiędzy językami C, C++, Java. Tablice, wskaźniki, referencje do lewej i prawej wartości. Kontenery (kolekcje) o charakterze tablicy. Typy wyliczeniowe, unie. Typy użytkownika - struktury, klasy. Hierarchie dostępu (private, protected, public).	2
W4	Funkcje oraz metody składowe klasy. Typy zwracane przez funkcje, rodzaje argumentów funkcji. Wskaźnik na funkcję oraz wskaźnik do metody składowej klasy. Funkcje statyczne.	2
W5	Dynamiczne tworzenie obiektów oraz ich usuwanie - porównanie technik w językach C, C++ i Java (malloc/free, new/delete, garbage collector). Wyciek pamięci - techniki zapobiegania i diagnozowania. Inteligentne wskaźniki.	2
W6	Zasady projektowania obiektowego. Pojęcie obiektu i klasy. Relacje pomiędzy obiektami - agregacja, dziedziczenie. Rodzaje dziedziczenia. Dziedziczenie wielokrotne w C++. Metody wirtualne i polimorfizm.	2
W7	Funkcje, biblioteki matematyczne i struktury danych dostępne w językach C, C++, Java, wspierające obliczenia numeryczne i inżynierskie. Generatory liczb losowych o różnych rozkładach.	2



WYKŁAD

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W8	Metody identyfikowania problemów w programach. Asercje i praca z debuggerem. Warunkowe fragmenty kodu. Profilowanie kodu w celu optymalizacji jego działania. Testy kodu.	1
	RAZEM	15

PROJEKT

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P1	Praktyczne przygotowanie i sprawdzenie środowiska pracy - wybranie edytorów kodu źródłowego, organizacja kodu, rozszerzenia plików źródłowych w językach C, C++, Java. Sprawdzenie kompilatorów (np. gcc, g++, javac z JDK), opcji kompilowania oraz uruchamiania kodu. Prosty program obsługujący strumienie wejścia i wyjścia. Argumenty programu.	3
P2	Program kalkulator, testujący operatory i funkcje matematyczne. Operacje wejścia i wyjścia - zasady konwersji pomiędzy typami (znakowym a całkowitym i zmiennoprzecinkowym). Dyskusja możliwych rozwiązań wizualizacji (grafika znakowa - ncurses, biblioteka Qt w C++, grafika w Java za pomocą biblioteki Swing).	4
P3	Program operujący na łańcuchach znakowych. Obsługa łańcuchów znakowych w języku C (funkcje z nagłówka string.h), typ string w języku C++ i String w Java oraz ich funkcjonalność.	4
P4	Program obliczający średnią, średnią ważoną, odchylenie standardowe. Wizualizacja rozkładu za pomocą znaków.	3
P5	Proste struktury danych - tablice. Tablice realizowane statycznie i dynamicznie. Tablice wielowymiarowe. Tablice realizowane przez kontenery (array, vector). Tablice heterogeniczne (tuple).	3
P6	Rekurencja - liczenie silni. Realizacja obliczenia silni podczas kompilacji (wyrażenie constexpr w C++11). Obliczanie wyznacznika macierzy rekurencyjną metodą rozwinięcia Laplace'a. Realizacja w C/C++ oraz jako applet Javy.	4
P7	Podstawowe typy abstrakcji danych: od struktury i unii do klasy. Tworzenie obiektów i relacji pomiędzy nimi. Kanoniczne metody składowe klas (w C++ konstruktory, konstruktory kopiujące i przenoszące, destruktor, operatory przypisania kopiujące i przenoszące). Metody jako sposób komunikacji między obiektami.	3
P8	Relacje obiektowe - dziedziczenie jednokrotne i metody wirtualne. Proste przykłady pokazujące zasady działania polimorfizmu w C++ i w Java.	2
P9	Dziedziczenie wielobazowe (C++) i problemy z tym związane. Problem bliźniąt i techniki zapobiegania konfliktom na przykładach fragmentów kodów.	2
P10	Wzorce projektowe - jako przykłady optymalnych rozwiązań określonych problemów w projektach informatycznych. Implementacja prostych wzorców (singleton, adapter, fasada) w językach C++ i Java.	2
P11	Wzorec "obserwator" - powiadamianie klientów o zdarzeniu. Implementacja w C++ i w Java na przykładzie urządzenia wykonującego pomiary. Klientem mogą być: wyświetlacz, pamięć magazynująca, inna maszyna czekająca na dane.	4
P12	Wzorec "dekorator" - dodawanie funkcjonalności lub cech do głównego obiektu. Implementacja w C++ i w Java na przykładzie systemu druku dokumentów, organizacji pracy kawiarni itp. Wzorec M-V-O w Java.	4
P13	Wzorec "fabryki abstrakcyjnej" - dostarczanie konkretnych obiektów poprzez zewnętrzną wydzieloną abstrakcję danych - implementacja w C++ i w Java. Metoda fabryczna. Metoda szablonowa.	3
P14	Przegląd i praktyczne korzyści z użytkowania kontenerów (kolekcji), iteratorów oraz wybranych algorytmów uogólnionych - nowoczesne techniki programowania.	2



PROJEKT

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P15	Przegląd i praktyczne przykłady bibliotek matematycznych oraz generatorów losowych dostępnych w C/C++11 oraz Java.	2
	RAZEM	45

7 METODY DYDAKTYCZNE

M1 Wykłady

M2 Ćwiczenia projektowe

M3 Prezentacje multimedialne

M4 Studium przypadku

M5 Praca w grupach

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	60
Konsultacje przedmiotowe	8
Egzaminy i zaliczenia w sesji	2
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	20
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	35
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	125
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	5

9 SPOSOBY OCENY

Nacisk położony na część praktyczną (poznawanie technik programowania oraz składni języka) z akcentem na konieczność samodzielnego dalszego kształcenia się i praktyki.

OCENA FORMUJĄCA

F1 Kolokwium

F2 Projekt indywidualny

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 1
---------------------	------------------------	-----------------------



NA OCENĘ 3	Student wie jak napisać, skompilować i uruchomić program w języku C, C++, Java. Nie pamięta niektórych składni w tych językach, ale zasadniczo daje sobie radę w pisaniu prostych programów oraz operacji wejścia / wyjścia. Czasem ma kłopoty z operacjami na plikach. Ma też kłopoty z metodami numerycznymi - proponowaniem rozwiązań prostych zagadnień.	projekt	Weryfikacja poprzez programowane wdrożenie konkretnych zadań, część z nich do zrealizowania w każdym z dyskutowanych języków, ale za pomocą innej składni. Wykazanie się umiejętnością kompilowania, konsolidacji i uruchamiania kodu wraz z konfigurowaniem odpowiednich ścieżek dostępu do bibliotek i innych składowych.
NA OCENĘ 4	Potrafi napisać, skompilować i uruchomić program w języku C, C++, Java. Czasem miewa wątpliwości co do niektórych struktur składniowych języka, czy priorytetów operatorów, ale zasadniczo powinien potrafić odszukać konieczną informację w materiałach źródłowych. Potrafi operować na strumieniach wejścia/ wyjścia, ale może w jednym z języków ma opanowane to lepiej, a w drugim gorzej.	projekt	
NA OCENĘ 5	Bez problemu potrafi napisać program w dowolnym z języków (C, C++, Java), wie jak go skompilować i jak uruchomić zarówno w systemie Windows jak i Linux. Zna typy, operatory, wyrażenia warunkowe i pętle, metody działania na strumieniach wejścia / wyjścia w tym zapisu i odczytu z plików. Umie wykonać działania na łańcuchach znakowych i typach string. Potrafi zaproponować rozwiązanie prostych zagadnień numerycznych.		
EFEKT KSZTAŁCENIA 2		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 2
NA OCENĘ 3	Student zna elementarne zasady działania na wskaźnikach i przez referencje, również w kontekście funkcji (tego co zwracają oraz jakie mogą przyjmować argumenty). Potrafi wykorzystać tablice, wykreować i zarządzać obiektami na stercie. Potrafi napisać klasę, ale nie zawsze zna szczegóły dotyczące jej kluczowych metod składowych, albo warunków, kiedy one są generowane automatycznie.	projekt	Stworzenie mini-projektu z kodem rozproszonym w wielu plikach, zwłaszcza w kontekście tworzenia zależności obiektowych (pisanie klas - abstrakcja, hermetyzacja i hierarchia danych).



NA OCENĘ 4	Student wie co to są wskaźniki i referencje i co za ich pomocą można zrobić. Potrafi tworzyć tablice statyczne i dynamiczne oraz nimi zarządzać. Nie zawsze pewnie odczytuje typy złożone (wskazujące na część tablicy). Wie co to jest i jak działa funkcja, ale pewne szczegóły (np. na temat nadawania wartości domyślnych w powtarzanych deklaracjach C++) są mu nieznane. Potrafi pisać funkcje i metody składowe klas. Może mieć pewne kłopoty z napisaniem i utrzymaniem projektu napisanego w wielu plikach (zwłaszcza w C i C++). Wie jak pisać klasy, nie zawsze jednak optymalnie je projektuje.		
NA OCENĘ 5	Student potrafi tworzyć tablice jedno- i wielowymiarowe, wyliczać ich indeksy. Zna typy częściowych tablic (bez podania wszystkich wymiarów). Użytkuje wskaźniki i referencje, rozumie co oznacza modyfikator const. Potrafi dynamicznie kreować (i jeśli trzeba - usuwać) obiekty i tablice. Potrafi napisać funkcję i metodę składową klasy, rozumie co może zwracać funkcja oraz jakie przyjmować argumenty. Potrafi napisać strukturę (C, C++), klasę (C++, Java). Potrafi zaprojektować fundamentalne metody dotyczące mechaniki życia obiektu (konstruktory, destruktory, operatory przypisania - oczywiście w zależności od języka w którym operuje i w którym takie metody występują lub nie występują). Potrafi napisać kod w wielu plikach (rozumie jak to się pisze w każdym z języków i jakie są wymagania nazewnictwa, czy wzgl. rozszerzeń plików) i go odpowiednio skompilować.		
EFEKT KSZTAŁCENIA 3		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 3
NA OCENĘ 3	Student zna podstawowe mechanizmy projektowania i programowania obiektowego. Potrafi odziedziczyć z typu bazowego i wykorzystać polimorfizm. Potrafi wskazać na różnice w C++ i Java. Potrafi elementarnie dziedziczyć też wielobazowo (C++) ale bez umiejętności rozwiązywania konfliktów potencjalnie z tego wynikających. Zna i korzysta z niektórych (najczęściej wykorzystywanych) kontenerów / kolekcji, ale często nie zna szczegółów składniowych i mechaniki iteratorów. Nie kojarzy algorytmów uogólnionych.	projekt	Na przykładzie programowanego studium przypadku student dyskutuje różne warianty rozwiązań oraz wykonuje ich implementację w C++ i w Java. Wykazuje się znajomością kontenerów i umie użyć odpowiedni rodzaj w zależności od potrzeby.



NA OCENĘ 4	Student potrafi napisać klasy bazowe i potomne, choć może nie radzi sobie z dziedziczeniem wielokrotnym (C++). Zna jednak i stosuje mechanizm polimorfizmu. Potrafi wskazać na optymalne rozwiązania na przykładach wzorców projektowych. Zna podstawowe kontenery i kolekcje i poruszać się po nich iteratorami. Potrafi również zastosować algorytm (C++) działający a kontenerze.		
NA OCENĘ 5	Student zna i stosuje zasady projektowania i programowania obiektowego (C++ i Java) oraz proceduralnego (C). Stosuje mechanizm dziedziczenia i polimorfizmu. Rozpoznaje zalety i wady dziedziczenia wielokrotnego (C++) w tym dziedziczenia wirtualnego. Wie co to jest typ abstrakcyjny. Potrafi wskazać na optymalne rozwiązania problemów projektowych w oparciu o wzorce projektowe i implementować je w C++ oraz w Java. Rozumie też na czym polega idiom programowania uogólnionego oraz sposoby jego realizacji poprzez kontenery i algorytmy. Zna rodzaje kontenerów (kolekcji). Zna przykładowe algorytmy.		
EFEKT KSZTAŁCENIA 4		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 4
NA OCENĘ 3	Student wykonuje operacje na bitach ale nie zawsze trafnie sam potrafi je wyliczyć. Raczej dobrze używa operatorów bitowych. Orientuje się w podstawowych możliwościach bibliotek matematycznych. Potrafi wykorzystać generator o rozkładzie płaskim do przeprowadzenia elementarnej symulacji.	projekt	Rozwiązywanie praktycznych ćwiczeń z programowania niskopoziomowego. Z drugiej strony wykazanie się pewną wiedzą ze statystyki, włączającą procesy działania generatorów losowych, implementowanie przykładów generatora o zadanym rozkładzie.
NA OCENĘ 4	Student potrafi wykonać działania niskiego poziomu (bitowe), nie myli operatorów bitowych z innymi operatorami. Orientuje się niezle w zakresie funkcji i algorytmów dostępnych w bibliotekach matematycznych. W szczególności potrafi napisać symulację prostego procesu statystycznego z użyciem generatora losowego oraz dostosować jego rozkład do własnych potrzeb.		



NA OCENĘ 5	Student wprawnie oblicza operacje bitowe w różnych systemach liczbowych. Zna podstawowe metody numeryczne stosowane w pracy inżyniera oraz biblioteki matematyczne dostarczane z językami. Potrafi wykorzystać generator liczb losowych do przeprowadzenia symulacji. Potrafi napisać generator o dowolnym rozkładzie i wie, że C++11 dostarcza takiej funkcjonalności w standardzie.		
------------------	---	--	--

OCENA DO INDEKSU (OCENA PODSUMOWUJĄCA)

Ważną częścią oceny jest egzamin, ale również aktywność i wykazywanie zainteresowania i postępów pod czas prac projektowych i ćwiczeniowych.

WARUNKI ZALICZENIA PRZEDMIOTU

- a Uzyskanie pozytywnej oceny z części projektowej oraz egzaminu końcowego. Ocena końcowa to średnia arytmetyczna uzyskanych ocen.

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKTY KSZTAŁCENIA DLA PRZEDMIOTU	ODNIESIENIE DO EFEKTÓW KIERUNKOWYCH	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	METODY DYDAKTYCZNE
EK1	MT_UO03, MT_W06	Cel1	W1, W2, P1, P2, P3, P4	M1, M2, M3
EK2	MT_UO03, MT_W06, MT_UB07	Cel1, Cel2	W3, W4, W5, P5, P6, P7	M1, M2, M3
EK3	MT_UO03, MT_UP03, MT_UB07	Cel2, Cel3	W1, W6, W7, P8, P9, P10, P11, P12, P13	M1, M2, M3, M4, M5
EK4	MT_UO03, MT_UP03, MT_W06, MT_UB07	Cel3, Cel4	W1, W7, W8, P14, P15	M1, M2, M3, M4, M5

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA:

- [1] Stephen Prata — *Język C. Szkoła programowania. Wydanie V*, Warszawa, 2006, Helion
- [2] Brian W. Kernighan, Dennis M. Ritchie — *Język ANSI C. Programowanie. Wydanie II*, Warszawa, 2010, Helion
- [3] K. N. King — *Język C. Nowoczesne programowanie. Wydanie II*, Warszawa, 2011, Helion



- [4] Jerzy Grębosz — *Symfonia C++ Standard*, Kraków, 2010, Editions 2000
- [5] Bruce Eckel, Chuck Allison — *Thinking in C++, tom I i II*, Warszawa, 2004, Helion
- [6] Nicholas A. Solter, Scott J. Kleper — *C++ Zaawansowane programowanie*, Warszawa, 2005, Helion
- [7] Bruce Eckel — *Thinking in Java. Edycja polska. Wydanie IV*, Warszawa, 2006, Helion
- [8] Cay S. Horstmann, Gary Cornell — *Java. Podstawy. Wydanie VIII*, Warszawa, 2008, Helion
- [9] Herbert Schildt — *Java. Kompendium programisty*, Warszawa, 2005, Helion

LITERATURA UZUPEŁNIAJĄCA:

- [1] John Viega, Matt Messier — *C i C++. Bezpieczne programowanie. Receptury*, Warszawa, 2005, Helion / O'Reilly
- [2] Marek Tłuczek — *Programowanie w języku C. Ćwiczenia praktyczne. Wydanie II*, Warszawa, 2011, Helion
- [3] Bjarne Stroustrup — *Programowanie. Teoria i praktyka z wykorzystaniem C++*, Warszawa, 2010, Helion
- [4] D. Ryan Stephens i inni — *C++ Receptury*, Warszawa, 2006, Helion / O'Reilly
- [5] Marcin Lis — *Praktyczny kurs Java. Wydanie III*, Warszawa, 2011, Helion
- [6] Joshua Bloch — *Java. Efektywne programowanie. Wydanie II*, Warszawa, 2009, Helion

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr Witold Przygoda (kontakt: witold.przygoda@gmail.com)

OSOBY PROWADZĄCE PRZEDMIOT

dr Witold Przygoda (kontakt: witold.przygoda@gmail.com)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)	(odpowiedzialny za przedmiot)	(kierownik zakładu)	(dyrektor instytutu)
---------------------	-------------------------------	---------------------	----------------------

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....