



### KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	<b>Z-ZIP1-U-508a</b>
	studia niestacjonarne:	<b>Z-ZIPN1-U-508a</b>
Nazwa przedmiotu	<b>Języki programowania – C++</b>	
Nazwa przedmiotu w języku angielskim	<b>Programming Languages – C++</b>	
Obowiązuje od roku akademickiego	<b>2019/2020</b>	

### USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	<b>ZARZĄDZANIE I INŻYNIERIA PRODUKCJI</b>
Poziom kształcenia	<b>I stopień</b>
Profil studiów	<b>Ogólnoakademicki</b>
Forma i tryb prowadzenia studiów	<b>Studia stacjonarne i niestacjonarne</b>
Zakres	<b>Wszystkie zakresy</b>
Jednostka prowadząca przedmiot	<b>Katedra Technologii Informatycznych</b>
Koordinator przedmiotu	<b>dr hab. Marzena Nowakowska, prof. PŚk</b>
Zatwierdził	<b>dr hab. inż. Artur Bartosik, prof. PŚk</b>

### OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	<b>Przedmiot kierunkowy</b>	
Status przedmiotu	<b>Wybieralny</b>	
Język prowadzenia zajęć	<b>Polski</b>	
Usytuowanie w planie studiów - semestr	studia stacjonarne	<b>Semestr V</b>
	studia niestacjonarne	<b>Semestr V</b>
Wymagania wstępne	<b>Podstawy informatyki, Informatyka - programowanie (...)</b>	
Egzamin (TAK/NIE)	<b>NIE</b>	
Liczba punktów ECTS	<b>3</b>	

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	studia stacjonarne:	<b>15</b>		<b>30</b>		
	studia niestacjonarne:	<b>9</b>		<b>18</b>		

## EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna i rozumie zasady funkcjonowania programów komputerowych i zarządzania nimi w środowisku Windows.	ZIP_W05
	W02	Student ma podstawową wiedzę na temat typów danych i doboru tych typów do rozwiązania określonego zadania programistycznego.	ZIP_W05
	W03	Student rozumie modułową budowę programu komputerowego oraz potrzebę wykorzystania w nim wbudowanych funkcji C++ w środowisku IDE i własnych funkcji programisty.	ZIP_W05
Umiejętności	U01	Student potrafi zaprojektować i zbudować aplikację okienkową, korzystając z gotowych obiektów systemu programistycznego C++ w IDE oraz własnych rozwiązań programistycznych.	ZIP_U07
	U02	Student potrafi dokonać analizy prostego kodu źródłowego oraz wprowadzić modyfikacje w treści istniejącego programu.	ZIP_U07
	U03	Student posiada umiejętności definiowania własnych funkcji programisty oraz wykorzystania ich w tworzonej przez siebie aplikacji komputerowej.	ZIP_U07
	U04	Student posiada umiejętności konstruowania algorytmów służących rozwiązaniu różnych zadań programistycznych zgodnie z zasadami logiki uniwersalnej.	ZIP_U07
Kompetencje społeczne	K01	Student rozumie potrzebę stałego stosowania i uzupełniania wiedzy z zakresu działania algorytmicznego.	ZIP_K01
	K02	Student jest gotów pracować samodzielnie i w grupie (przyjmując w niej różne role).	ZIP_K04

## TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	<ol style="list-style-type: none"><li>1. Środowisko pracy IDE. Struktura programu w języku C++. Struktura aplikacji C++ w IDE. Tworzenie kodu wynikowego. Paleta komponentów, komunikacja z użytkownikiem.</li><li>2. Podstawowe instrukcje języka. Typy proste. Konwersja typów. Wybrane operatory, ich priorytet i łączność.</li><li>3. Algorytmy przetwarzania iteracyjnego. Tablice i instrukcje pętli.</li><li>4. Typy wskaźnikowe. Operatory wskazania i wyłuskania. Arytmetyka</li><li>5. adresowa. Praca z tablicami. Sterowanie w programie z wykorzystaniem instrukcji pętli.</li><li>6. Definiowanie funkcji i przekazywanie parametrów. Elementy programowania obiektowego na przykładzie komponentów dostępnych w IDE. Właściwości i metody klas jako narzędzia działania na obiektach.</li><li>7. Klasa łańcuchów tekstowych; właściwości i metody. Przetwarzanie tekstów.</li><li>8. Organizacja dostępu do pliku tekstowego z wykorzystaniem komponentów dialogowych. Współpraca aplikacji z plikiem tekstowym. Transmisja danych między aplikacją i plikiem tekstowym.</li></ol>

laboratorium	<ol style="list-style-type: none"> <li>1. Struktura projektu w środowisku IDE dla C++. Komunikacja aplikacji z użytkownikiem. Kolekcja obiektów standardowych.</li> <li>2. Konwersja typów i operacje arytmetyczne. Kolejność działań w programie. Sterowanie w programie. Wprowadzanie danych i wyprowadzanie wyników.</li> <li>3. Algorytmy obliczeniowe. Przetwarzanie iteracyjne danych numerycznych. Implementacja algorytmów obliczeniowych w aplikacji C++ utworzonej w IDE. Tablice i instrukcje pętli: wyznaczanie statystyk z tablic numerycznych.</li> <li>4. Przetwarzanie iteracyjne tablic z wykorzystaniem klasy typów tekstowych.</li> <li>5. Funkcja użytkownika jako składowa klasy i jako funkcja zewnętrzna. Zmienne globalne. Przekazywanie parametrów funkcji.</li> <li>6. Przetwarzanie ciągów znaków. Iteracyjne przetwarzanie tekstów: statystyki, wyszukiwanie i modyfikowanie tekstów.</li> <li>7. Organizacja dostępu do plików tekstowych. Transmisja danych między pamięcią dyskową i pamięcią operacyjną (struktury tablicowe i wizualne komponenty formularza). Tworzenie aplikacji współpracującej z plikiem tekstowym.</li> </ol>
--------------	---

### **METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ**

Symbol efektu	Metody sprawdzania efektów kształcenia <i>(zaznaczyć X)</i>					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X			X
W02			X			X
W03			X			X
U01			X			X
U02			X			X
U03			X			X
U04			X			X
K01						X
K02						X

### **FORMA I WARUNKI ZALICZENIA**

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów ze sprawdzianu końcowego.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów z kolokwίων w trakcie zajęć.

## NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jednostka
		studia stacjonarne					studia niestacjonarne					
		W	C	L	P	S	W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	15		30			9		18			h
2.	Inne (konsultacje, egzamin)	2		2			2		2			h
3.	<b>Razem przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>49</b>					<b>31</b>					h
4.	<b>Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>2,0</b>					<b>1,2</b>					ECTS
5.	<b>Liczba godzin samodzielnej pracy studenta</b>	<b>26</b>					<b>44</b>					h
6.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy</b>	<b>1,0</b>					<b>1,8</b>					ECTS
7.	<b>Nakład pracy związany z zajęciami o charakterze praktycznym</b>	<b>50</b>					<b>50</b>					h
8.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym</b>	<b>2,0</b>					<b>2,0</b>					ECTS
9.	<b>Sumaryczne obciążenie pracą studenta</b>	<b>75</b>					<b>75</b>					h
10.	<b>Punkty ECTS za moduł</b> <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	<b>3</b>										ECTS

## LITERATURA

1. Allain A. (2014), *C++ przewodnik dla początkujących*, Helion, Gliwice.
2. Čukić I. (2019), *Programowanie funkcyjne w C++: tworzenie lepszych aplikacji*, Helion, Gliwice.
3. Grębosz J. (2018), *Opus magnum C++11. Programowanie w języku C++*, Helion, Gliwice.
4. Stroustrup B. (2014), *Język C++. Kompendium wiedzy*, Helion, Gliwice.
5. Swart B., Cashman M., Gustavson P., Hollingworth J., *Borland C++Builder 6 Developer's Guide*, SAMS, dostępne ze strony: [http://140.129.118.16/~richwang/ktlan/BCPPB6\\_Book\\_unlocked.pdf](http://140.129.118.16/~richwang/ktlan/BCPPB6_Book_unlocked.pdf).
6. Wróblewski P. (2019), *Algorytmy, struktury danych i techniki programowania*, Helion, Gliwice.