

Wymagania edukacyjne z informatyki dla klasy 8 szkoły podstawowej.

Tytuł w podręczniku	Numer i temat lekcji	Wymagania na ocenę dopuszczającą. Uczeń:	Wymagania na ocenę dostateczną. Uczeń:	Wymagania na ocenę dobrą. Uczeń:	Wymagania na ocenę bardzo dobrą. Uczeń:	Wymagania na ocenę celującą. Uczeń:
OCENA ŚRÓDROCZNA						
DZIAŁ 1. Arkusz kalkulacyjny						
1.1. Formuły i adresowanie względne w arkuszu kalkulacyjnym	1. i 2. Formuły i adresowanie względne w arkuszu kalkulacyjnym	<ul style="list-style-type: none"> • omawia zastosowanie oraz budowę arkusza kalkulacyjnego • określa adres komórki • wprowadza dane różnego rodzaju do komórek arkusza kalkulacyjnego • formatuje zawartość komórek (wyrównanie tekstu oraz wygląd czcionki) 	<ul style="list-style-type: none"> • określa zasady wprowadzania danych do komórek arkusza kalkulacyjnego • dodaje i usuwa wiersze oraz kolumny w tabeli 	<ul style="list-style-type: none"> • tworzy proste formuły obliczeniowe • wyjaśnia, czym jest adres względny 	<ul style="list-style-type: none"> • kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne 	<ul style="list-style-type: none"> • samodzielnie tworzy i kopiuje skomplikowane formuły obliczeniowe

1.2. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	3. i 4. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	<ul style="list-style-type: none"> • rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym 	<ul style="list-style-type: none"> • stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora 	<ul style="list-style-type: none"> • wykorzystuje funkcję JEŻELI do tworzenia algorytmów z warunkami w arkuszu kalkulacyjnym • ustawia format danych komórki odpowiadający jej zawartości • w formułach stosuje adresowanie względne, bezwzględne i mieszane 	<ul style="list-style-type: none"> • korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje • stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych 	<ul style="list-style-type: none"> • stosuje zaawansowane funkcje arkusza w tabelach tworzonych na własne potrzeby
1.3. Przedstawianie danych na wykresie	5. i 6. Przedstawianie danych na wykresie	<ul style="list-style-type: none"> • wstawia wykres do arkusza kalkulacyjnego 	<ul style="list-style-type: none"> • omawia i modyfikuje poszczególne elementy wykresu 	<ul style="list-style-type: none"> • dobiera odpowiedni wykres do rodzaju danych 	<ul style="list-style-type: none"> • tworzy wykres dla więcej niż jednej serii danych 	<ul style="list-style-type: none"> • tworzy rozbudowane wykresy dla wielu serii danych
1.4. Zastosowania arkusza kalkulacyjnego	7. i 8. Zastosowania arkusza kalkulacyjnego	<ul style="list-style-type: none"> • korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków 	<ul style="list-style-type: none"> • zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie 	<ul style="list-style-type: none"> • sortuje oraz filtruje dane w arkuszu kalkulacyjnym 	<ul style="list-style-type: none"> • tworzy prosty model (na przykładzie rzutu sześcienną kostką do gry) w arkuszu kalkulacyjnym • stosuje filtry niestandardowe 	<ul style="list-style-type: none"> • przygotowuje rozbudowane arkusze kalkulacyjne korzysta z arkusza kalkulacyjnego do analizowania doświadczeń

						z innych przedmiotów
DZIAŁ 2. Algorytmika i projektowanie						
2.1. Wprowadzenie do programowania w języku C++	9. i 10. Wprowadzenie do programowania w języku C++	<ul style="list-style-type: none"> definiuje pojęcia: algorytm, program, programowanie podaje kilka sposobów przedstawienia algorytmu 	<ul style="list-style-type: none"> wymienia różne sposoby przedstawienia algorytmu: opis słowny, schemat blokowy, lista kroków poprawnie formułuje problem do rozwiązania wyjaśnia różnice między interaktywnym a skryptowym trybem pracy stosuje odpowiednie polecenie języka C++, aby wyświetlić tekst na ekranie omawia różnice pomiędzy kodem źródłowym a kodem wynikowym tłumaczy, czym jest środowisko programistyczne 	<ul style="list-style-type: none"> wymienia przykładowe środowiska programistyczne wyjaśnia, czym jest specyfikacja problemu opisuje etapy rozwiązywania problemów opisuje etapy powstawania programu komputerowego zapisuje proste polecenia języka C++ 	<ul style="list-style-type: none"> pisze proste programy w trybie skryptowym języka C++ 	<ul style="list-style-type: none"> zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności

2.2. Piszemy programy w języku C++	11. i 12. Piszemy programy w języku C++	<ul style="list-style-type: none"> • tłumaczy, do czego używa się zmiennych w programach • pisze proste programy w trybie skryptowym języka C++ z wykorzystaniem zmiennych 	<ul style="list-style-type: none"> • wykonuje obliczenia w języku C++ • omawia działanie operatorów arytmetycznych • stosuje tablice w języku C++ oraz operatory logiczne 	<ul style="list-style-type: none"> • wykorzystuje instrukcję warunkową if oraz else w programach • wykorzystuje iterację w konstruowanych algorytmach • wykorzystuje w programach instrukcję iteracyjną for • definiuje funkcje w języku C++ i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości 	<ul style="list-style-type: none"> • buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów • konstruuje złożone sytuacje warunkowe (wiele warunków) w algorytmach • pisze programy zawierające instrukcje warunkowe, pętle oraz funkcje • wyjaśnia, jakie błędy zwraca interpreter • czyta kod źródłowy i opisuje jego działanie 	<ul style="list-style-type: none"> • pisze programy w języku C++ do rozwiązywanie zadań matematycznych • tworzy program składający się z kilku funkcji wywoływanych w programie głównym
2.3. Wprowadzenie do programowania w języku Python	13. i 14. Wprowadzenie do programowania w języku Python	<ul style="list-style-type: none"> • definiuje pojęcia: algorytm, program, programowanie • podaje kilka sposobów przedstawienia algorytmu 	<ul style="list-style-type: none"> • wymienia różne sposoby przedstawienia algorytmu: opis słowny, schemat blokowy, lista kroków 	<ul style="list-style-type: none"> • wymienia przykładowe środowiska programistyczne • wyjaśnia, czym jest specyfikacja problemu 	<ul style="list-style-type: none"> • pisze proste programy w trybie skryptowym języka Python 	<ul style="list-style-type: none"> • zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności

			<ul style="list-style-type: none"> • poprawnie formułuje problem do rozwiązania • wyjaśnia różnice między interaktywnym a skryptowym trybem pracy • stosuje odpowiednie polecenie języka Python, aby wyświetlić tekst na ekranie • omawia różnice pomiędzy kodem źródłowym a kodem wynikowym • tłumaczy, czym jest środowisko programistyczne 	<ul style="list-style-type: none"> • opisuje etapy rozwiązywania problemów • opisuje etapy powstawania programu komputerowego • zapisuje proste polecenia języka Python • 		
2.4. Piszemy programy w języku Python	15. i 16. Piszemy programy w języku Python	<ul style="list-style-type: none"> • tłumaczy, do czego używa się zmiennych w programach • pisze proste programy w trybie skryptowym języka Python z wykorzystaniem zmiennych 	<ul style="list-style-type: none"> • wykonuje obliczenia w języku Python • omawia działanie operatorów arytmetycznych • stosuje listy w języku Python oraz operatory logiczne 	<ul style="list-style-type: none"> • wykorzystuje instrukcję warunkową if oraz if else w programach • wykorzystuje iterację w konstruowanych algorytmach • wykorzystuje w programach 	<ul style="list-style-type: none"> • buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów • konstruuje złożone sytuacje warunkowe 	<ul style="list-style-type: none"> • pisze programy w języku Python do rozwiązywania zadań matematycznych • tworzy program składający się z kilku funkcji wywoływanych

				instrukcję iteracyjną for • definiuje funkcje w języku Python i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości	(wiele warunków) w algorytmach • pisze programy zawierające instrukcje warunkowe, pętle oraz funkcje • wyjaśnia, jakie błędy zwraca interpreter • czyta kod źródłowy i opisuje jego działanie	w programie głównym
OCENA KOŃCOWOROCZNA						
2.5. Algorytmy wyszukiwania	17. i 18. Algorytmy wyszukiwania	• wyjaśnia potrzebę wyszukiwania informacji w zbiorze • określa różnice między wyszukiwaniem w zbiorach uporządkowanym i nieuporządkowanym • sprawdza działanie programów wyszukujących element w zbiorze	• zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym, w tym elemencie największego i najmniejszego • zapisuje algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia • implementuje grę w zgadywanie liczby	• implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym • omawia funkcje zastosowane w realizacji algorytmu wyszukiwania metodą połowienia • implementuje algorytm wyszukiwania	• samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze metodą połowienia, w tym elemencie największego i najmniejszego • implementuje algorytm wyszukiwania elementu	• samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania

				największej wartości w zbiorze	w zbiorze metodą połowienia	
2.6. Algorytmy porządkowania	19. i 20. Algorytmy porządkowania	<ul style="list-style-type: none"> • wyjaśnia potrzebę porządkowania danych • sprawdza działanie programu sortującego dla różnych danych 	<ul style="list-style-type: none"> • zapisuje w wybranej formie algorytm porządkowania metodami przez wybieranie oraz przez zliczanie • omawia implementację algorytmu sortowania przez wybieranie • stosuje pętle zagnieżdżone i wyjaśnia, jak działają 	<ul style="list-style-type: none"> • omawia implementację algorytmu sortowania przez zliczanie • omawia funkcje zastosowane w kodzie źródłowym algorytmów sortowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> • implementuje algorytmy porządkowania metodami przez wybieranie oraz przez zliczanie • wprowadza modyfikacje w implementacji algorytmów porządkowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> • samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez wybieranie, metodą przez zliczanie
DZIAŁ 3. Internet						
3.1. Tworzenie strony internetowej z wykorzystaniem języka HTML.	21. i 22. Wprowadzenie do znaczników języka HTML, tworzenie własnej strony internetowej w języku HTML.	• Tworzy prostą stronę internetową w języku HTML, wykorzystując edytor tekstu.	• Zapisuje utworzoną stronę internetową w formacie HTML.	• Omawia zasady projektowania stron internetowych, wyjaśnia działanie hiperłączy.	• Modyfikuje kod utworzonej strony internetowej, wyszukuje błędy w utworzonym kodzie.	• Tworzy hiperłącza w budowanej stronie internetowej, dodaje tło do tworzonej strony.
3.2. Praca w chmurze.	23. i 24. Wspólny projekt internetowy.	• Umieszcza pliki w chmurze.	• Udostępnia pliki umieszczone w chmurze innym,	• Rozdziela pomiędzy członków grupy zadania niezbędne	• Krytycznie ocenia wartość informacji	• Podczas pracy nad projektem wykazuje się

			współpracuje z innymi podczas wykonywania wspólnego projektu.	do wykonania projektu.	znalezionych w internecie – weryfikuje je w różnych źródłach.	wysokim poziomem estetyki i kreatywności.
DZIAŁ 4. Projekty						
4.1. Dokumentacja szkolnej imprezy sportowej	25. i 26. Dokumentacja szkolnej imprezy sportowej	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone mu zadania o niewielkim stopniu trudności 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej • wprowadza dane do zaprojektowanych tabel 	<ul style="list-style-type: none"> • przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy • współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, przygotowuje zestawienia, drukuje wyniki • współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, tworzy zestawienia zawierające zaawansowane formuły, wykresy oraz elementy graficzne • współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera
4.2. Sterowanie obiektem na ekranie	27. i 28. Sterowanie obiektem na ekranie	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności 	<ul style="list-style-type: none"> • bierze udział w pracach nad wypracowaniem koncepcji gry 	<ul style="list-style-type: none"> • programuje wybrane funkcje i elementy gry • opracowuje opis gry 	<ul style="list-style-type: none"> • implementuje i optymalizuje kod źródłowy gry, korzystając 	<ul style="list-style-type: none"> • rozbudowuje grę o nowe elementy • współpracuje w grupie podczas pracy nad

		<ul style="list-style-type: none"> • testuje grę na różnych etapach • współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> • współpracuje w grupie podczas pracy nad projektem 		z wypracowanych założeń	projektem, przyjmuje funkcję lidera
4.3. Historia i rozwój informatyki	29. i 30. Historia i rozwój informatyki	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności – znalezienie informacji w internecie, umieszczenie ich w chmurze 	<ul style="list-style-type: none"> • współpracuje z innymi podczas pracy nad projektem • analizuje zebrane dane • tworzy projekt prezentacji multimedialnej 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania • tworzy prezentację wg projektu zaakceptowanego przez zespół 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu • analizuje i weryfikuje pod względem merytorycznym i technicznym przygotowaną prezentację 	<ul style="list-style-type: none"> • współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera • wzbogaca prezentację o elementy podnoszące jej walory estetyczne i merytoryczne
4.4. Informatyka w moim przyszłym życiu	31. Informatyka w moim przyszłym życiu	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności • bierze aktywny udział w dyskusji nad wyborem atrakcyjnego zawodu wymagającego kompetencji informatycznych 	<ul style="list-style-type: none"> • gromadzi informacje dotyczące wybranych zawodów, umieszcza je w zaprojektowanych tabelach i dokumentach tekstowych 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu • projektuje tabele do zapisywania informacji o zawodach • weryfikuje i formatuje przygotowane dokumenty tekstowe 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, weryfikuje opracowane treści i łączy wszystkie dokumenty w całość 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, przyjmuje rolę lidera • podczas dyskusji przyjmuje funkcję moderatora

