

Wymagania edukacyjne z informatyki dla klasy drugiej

Osiągnięcia uczniów				
Na ocenę dopuszczający uczeń:	Na ocenę dostateczny uczeń:	Na ocenę dobry uczeń:	Na ocenę bardzo dobry uczeń:	Na ocenę celujący uczeń:
Rozdział I. Tworzenie treści internetowych				
– rozumie i rozróżnia podstawowe elementy składni języka HTML oraz zna strukturę strony internetowej, – rozumie model działania aplikacji do wizualnego generowania stron i systemów zarządzania treścią CMS, – uczestniczy w pracach zespołu i rozumie, jakie zadania wykonują poszczególni członkowie	– potrafi w prostym narzędziu tekstowym poprawnie zapisać podstawowe znaczniki, w tym stosować – potrafi przetestować działanie napisanego kodu strony WWW w przeglądarce internetowej, – potrafi zbudować prostą stronę w aplikacji do wizualnego generowania stron,	– potrafi weryfikować legalność pozyskiwanych materiałów dla swojej strony internetowej. – potrafi tworzyć i stosować arkusze stylów, – testuje działanie strony WWW na różnych urządzeniach (komputer, smartfon), – potrafi przetestować w przeglądarce internetowej stronę WWW zbudowaną za pomocą aplikacji, – potrafi współdziałać w zespole w zakresie projektowania i tworzenia strony internetowej,	– rozumie pojęcia tabeli, listy i stylów oraz potrafi modyfikować przygotowane wcześniej kody źródłowe zawierające wskazane pojęcia, – korzysta z legalnych źródeł do pozyskiwania treści i materiałów dla tworzonych stron WWW. – biegle potrafi tworzyć i modyfikować znaczniki z rozbudowanymi atrybutami, – potrafi stosować grafikę, tabele i inne elementy w odniesieniu do responsywności strony internetowej,	– potrafi stworzyć złożoną stronę, sterowaną zaprojektowanym menu i ją opublikować oraz przetestować na różnych urządzeniach (komputer, smartfon), – dba o identyfikację wizualną, korzysta z narzędzi graficznych i multimedialnych do wzbogacania treści, – tworzy podcasty i publikacje wideo na wybrane tematy, wymagające dużego nakładu pracy (np. promocja czy jubileusz szkoły, szkolny festiwal kultury lub nauki) lub korzysta

	– potrafi wykonać proste zadanie w zakresie przygotowania treści i zadanie techniczne w zakresie ich publikacji,		– rozumie, na czym polega zaawansowane zarządzanie treścią w systemach CMS, – opracowuje interesujące treści internetowe dostosowane do potrzeb potencjalnych odbiorców, wykorzystując zasadę 5W, – montuje materiały wykorzystując specjalistyczne oprogramowanie (np. Stream z pakietu Office 365), – występuje przed kamerą i mikrofonem, przekazuje treści w sposób atrakcyjny dla odbiorców, utrzymuje ich uwagę. – dba o ochronę prawa autorskiego publikowanych treści. – skutecznie weryfikuje możliwości wykorzystania materiałów pod kątem praw autorskich, ochrony	z zaawansowanych narzędzi. – wykazuje się kreatywnością w zakresie projektowania strony internetowej, – potrafi wykonać zaawansowane zadania techniczne,
--	--	--	--	--

			informacji oraz danych osobowych.	
Rozdział II. Relacyjne bazy danych				
– wyjaśnia podstawowe pojęcia związane z bazami danych: tabela, atrybut, rekord, pole, klucz główny, klucz obcy, relacja, – zarządza danymi w bazie danych w programie MS Access,	– wymienia różne zastosowania baz danych, – tworzy tabele i definiuje relacje między nimi	– projektuje nieduże bazy danych. – modyfikuje dane zawarte w bazie danych, dodaje nowe atrybuty i określa typ danych w Widoku projektu,	– projektuje rozbudowane bazy danych. – podaje przykłady baz danych, w których podczas projektowania należy uwzględnić relacje typu: jeden do jednego, jeden do wielu, wiele do wielu, – tworzy maski wprowadzenia, – tworzy kwerendy w programie MS Access, – modyfikuje bazy danych, poszerzając ich zastosowania.	– tworzy rozbudowane bazy danych, zawierające relacje typu: jeden do jednego, jeden do wielu, wiele do wielu,
Rozdział III. Algorytmika i programowanie w języku Python				
– wyjaśnia podstawowe pojęcia: algorytm, lista kroków, kod źródłowy, kod wynikowy, translator, kompilator, interpreter, słowa	– pisze programy rozwiązujące zadania matematyczne i fizyczne oraz problemy z napisami.	– wymienia cechy poprawnego algorytmu, tworzy proste algorytmy, – wyjaśnia na przykładzie pojęcie specyfikacji problemu,	– wymienia zasady tworzenia kodu źródłowego w wybranym języku programowania, – zapisuje algorytm w postaci kodu źródłowego,	– stosuje w swoich programach zagnieżdżone instrukcje warunkowe, – tworzy algorytmy konwersji między

kluczowe, blok instrukcji, zmienna, operator przypisania, argument, – wyjaśnia pojęcia: system pozycyjny, podstawa systemu liczbowego, dzielenie całkowite, reszta z dzielenia, – definiuje liczby złożone i liczby pierwsze oraz podstawowe twierdzenie arytmetyki, – podaje przykłady użycia liczb pierwszych, – omawia rozkład liczb na czynniki pierwsze, – definiuje pojęcia: porządkowanie (sortowanie), wyszukiwanie sekwencyjne i wyszukiwanie binarne (połówkowe),	– bada podzielność wybranych liczb, programując poznane algorytmy w wybranym języku, – tworzy samodzielnie program sortujący z użyciem wartownika, – wykonuje zadania o podwyższonym stopniu trudności, – stosuje algorytmy sortowania o mniejszej złożoności czasowej (szybkie, przez scalanie).	– stosuje podstawowe konstrukcje wybranego języka programowania: instrukcje wejścia i wyjścia, instrukcje warunkowe oraz pętle, – zapisuje liczby w systemie liczbowym: dziesiętnym, binarnym, ósemkowym i szesnastkowym, – wyjaśnia praktyczne znaczenie liczb pierwszych w informatyce. – wskazuje operacje kluczowe w algorytmach sortowania,	uruchamia zapisany kod źródłowy, – wyjaśnia pojęcie składni i błędu składniowego, znajduje i poprawia błędy w kodzie źródłowym programu. – tworzy samodzielnie programy, wykorzystując poznane instrukcje wybranego języka programowania, – omawia algorytm konwersji liczb między systemami dziesiętnym i binarnym, – zapisuje algorytm konwersji między systemami liczbowymi w postaci programu komputerowego – programuje algorytmy konwersji między różnymi systemami liczbowymi, stosując strukturę listy. – stosuje operację dzielenia całkowitego	różnymi systemami liczbowymi, – tworzy samodzielnie programy dla poznanych algorytmów,
--	--	---	--	---

– wyjaśnia znaczenie uporządkowania danych w procesie wyszukiwania,			w rozwiązywaniu problemów, – pisze programy wykorzystujące poznane rodzaje liczb pierwszych, – grupuje instrukcje w funkcje i wyjaśnia cel stosowania funkcji. – omawia algorytmy sprawdzające podzielność liczb, – wykorzystuje strukturalne typy danych (listy) do przechowywania danych, stosuje pętle zagnieżdżone, – omawia oraz implementuje algorytm sortowania bąbelkowego (prostej zamiany) i przez wstawianie w języku Python, zarówno nierosnąco, jak i niemalejąco, szacuje liczbę porównań oraz zamian w każdym z nich.	
---	--	--	---	--