

Plan wynikowy z wymaganiami edukacyjnymi z ***dodatkowych zajęć uzupełniających z zastosowań informatyki***
realizowanych w klasie 2 i 3 liceum ogólnokształcącego
w Zespole Szkół Samorządowych w Ełku

tematyka zajęć	ocena				
	dopuszczająca	dostateczna	dobra	bardzo dobra	celująca
	uczeń				
KLASA 2					
praca w systemie operacyjnym	- przestrzega normy i zasady BHP oraz zachowuje dbałość o stanowisko pracy - przestrzega regulamin pracowni komputerowej oraz norm współpracy w grupie - loguje się do systemu w środowisku graficznym - pracuje z powłoką graficzną systemu operacyjnego - wykonuje operacje na plikach i katalogach w środowisku graficznym (tworzy pliki i katalogi, przemieszcza się wewnątrz katalogów, wyświetla listę plików i katalogów, kopiuje, przenosi i usuwa pliki i katalogi) - poprawnie kończy pracę z systemem operacyjnym	- zna ideę oprogramowania wolnego i otwartego - wymienia warstwy systemu operacyjnego - zna i obsługuje programy wykonujące podstawowe zadania w środowisku graficznym - uruchamia programy, - przegląda i edytuje pliki tekstowe - przegląda pliki graficzne, - odtwarza i nagrywa dźwięki, -odtwarza filmy - rozpoznaje podstawowe typy plików i kojarzy z programem - wyszukuje i porównuje pliki w trybie graficznym	- loguje się do systemu w środowisku tekstowym - pracuje w trybie tekstowym systemu operacyjnego - korzysta z pomocy i podręcznika systemowego - wykonuje operacje na plikach i katalogach w trybie tekstowym (tworzy pliki i katalogi, przemieszcza się wewnątrz katalogów, wyświetla listę plików i katalogów, kopiuje, przenosi i usuwa pliki i katalogi) korzystając ze specjalistycznego programu	- charakteryzuje warstwy systemu operacyjnego - zna i wykonuje podstawowe polecenia w trybie tekstowym (np. wyświetlanie nazwy konta użytkownika, listy aktywnych kont, informacji o aktywnych użytkownikach, informacji o logowaniach użytkownika, informacji o procesach) - przegląda historię poleceń - wykonuje operacje na plikach i katalogach w trybie tekstowym	- rozróżnia różne sposoby licencjonowania - przekierowuje standardowe strumienie danych - charakteryzuje i rozpoznaje podstawowe systemy plików - wyszukuje i porównuje pliki w trybie tekstowym - aktualizuje system operacyjny i instaluje oprogramowanie dodatkowe - zabezpiecza komputer przed niepożądanym dostępem i przed wirusami

praca w sieci	- wymienia urządzenia sieciowe (np. karta sieciowa, media, przełącznik, koncentrator, most, router) - pracuje w sieci lokalnej i rozległej - rozumie zależność serwer - klient - przegląda strony WWW wysyła i odbiera pocztę elektroniczną - pracuje z klientem FTP - zachowuje podstawowe zasady bezpieczeństwa w sieci	- sprawdza konfigurację sieciową w trybie graficznym - charakteryzuje podstawowe rodzaje sieci i topologie sieciowe - omawia rolę urządzeń sieciowych - zna i rozumie podstawowe zagrożenia w sieci - przestrzega zasady etyczne i moralne w pracy z komputerem i w sieci komputerowej w tym netykietę	- stosuje polecenia sieciowe (np. ping, ifconfig, netstat, nslookup, traceroute, route) - uzyskuje dostęp do konta sieciowego za pomocą różnych usług (FTP, SSH, WWW) - pracuje zdalnie na komputerze sieciowym - wymienia pliki w sieci	- sprawdza konfigurację sieciową w trybie tekstowym - charakteryzuje podstawowe protokoły sieciowe - rozumie pojęcia serwer, serwer internetowy, klient sieciowy - odczytuje adres IP i MAC oraz maskę sieciową, klasy adresów	- posiada rozeznanie wśród nowoczesnych technologii komputerowych oraz świadomość zagrożeń i korzyści jakie niosą ze sobą
praca z edytorami grafiki rastrowej i wektorowej	- zna podstawowe formaty plików graficznych stosowane w dokumentach HTML	- rozróżnia typy plików graficznych - pracuje w edytorze grafiki rastrowej i wektorowej - przygotowuje grafikę na potrzeby dokumentów HTML	- wykorzystuje podstawowe narzędzia edytorów grafiki rastrowej i wektorowej	- eksportuje i importuje pliki graficzne - dokonuje transformacji obrazów	

Tworzenie dokumentów HTML z wykorzystaniem CSS	- tworzy dokumenty HTML w oparciu o edytor typu WYSIWIG lub tekstowy - eksportuje dokumenty do postaci HTML - pracuje z tekstowym edytorem HTML - stosuje podstawowe znaczniki HTML - publikuje dokumenty HTML za pomocą usługi FTP - uzyskuje dostęp do dokumentów HTML za pomocą przeglądarki internetowej	- rozwija umiejętność planowania pracy - przestrzega prawo, w tym prawo autorskie i szanuje własność intelektualną - zna i tworzy strukturę dokumentu HTML - poprawnie dołącza pliki CSS - stosuje podstawowe znaczniki HTML - definiuje odsyłacze wewnętrzne i zewnętrzne - wstawia grafikę i definiuje jej położenie i właściwości otaczania - tworzy i formatuje listy wypunktowane i uporządkowane - wstawia proste tabele	- formatuje i określa elementy strony WWW za pomocą stylów - definiuje selektory - tworzy i formatuje tabele (wstawia wiersze i kolumny, formatuje i ich zawartość, określa szerokość tabeli, łączy komórki, zmienia kolor tła)	- tworzy i dołącza zewnętrzne i wewnętrzne arkusze stylów oraz stosuje lokalnie style - tworzy formularze i organizuje i formatuje ich elementy (pola tekstowe, przyciski opcji, rozwijalne listy, pola wyboru, obszary tekstowe, pola ukryte)	- formatuje i określa wygląd strony za pomocą stylów (wybór czcionki, czcionki alternatywne kursywa, pogrubienie wielkość czcionki, kolor tekstu, kolor tła, wyrównanie tekstu, określanie wysokości i szerokości elementu, marginesów wokół elementu, rozmieszczanie i pozycjonowanie elementów) - wstawia skrypty języka programowania - definiuje prosty skrypt przetwarzania formularza
---	---	--	---	---	---

KLASA 3

Podstawy algorytmiki	- wie, co to jest algorytm - określa dane i wyniki do zadania - zna podstawowe zasady graficznego prezentowania algorytmów, podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym - potrafi narysować schemat blokowy algorytmu liniowego- określa sytuacje warunkowe - wie na czym polega powtarzanie tych samych operacji - potrafi ocenić poprawność algorytmu na podstawie analizy przebiegu algorytmu dla przykładowych danych	- podaje przykłady działań, które uważa za algorytmy - zna pojęcie specyfikacji zadania - zna wybrane sposoby prezentacji algorytmów - przedstawia algorytm w postaci listy kroków - tworzy schemat blokowy z warunkiem prostym i pętlą - obsługuje program do symulacji działania algorytmu w postaci schematu blokowego - odróżnia algorytmy liniowe i z rozgałęzieniami - rozumie pojęcie iteracji - omawia algorytm porządkowania metodą przez wybór na praktycznym przykładzie - określa liczbę prostych działań zawartych w algorytmie	- określa zależności między problemem, algorytmem a programem komputerowym - podaje dokładną specyfikację wybranego zadania - testuje algorytmy za pomocą programu symulującego jego działanie - zna algorytm Euklidesa, sortowania metodą przez wybór, znajdowania największego (najmniejszego) elementu w zbiorze - zna rekurencyjną realizację wybranego algorytmu	- zapisuje dowolny algorytm w wybranej przez siebie postaci - przeprowadza analizę poprawności konstrukcji schematu blokowego - stosuje swobodnie program do symulacji działania algorytmu na podstawie jego schematu blokowego - zna algorytm generowania liczb Fibonacciego, schemat Hornera - wskazuje różnicę między iteracją a rekurencją - zastępuje algorytmy rekurencyjne iteracyjnymi i odwrotnie - potrafi porównać złożoność algorytmów - rozumie pojęcie algorytmu uniwersalnego	- samodzielnie wyszukuje algorytmy w różnych dziedzinach szkolnych i sporządza ich zapisy - stosuje poznane techniki algorytmiczne w rozwiązaniu innych zadań - rozumie rekurencję - ocenia kiedy stosować iterację a kiedy rekurencję
----------------------	---	---	---	---	---

Podstawy programowania	- zna klasyfikację języków programowania - zna strukturę programu komputerowego oraz najważniejsze elementy języka - obsługuje środowisko programistyczne - zapisuje prosty algorytm liniowy w języku programowania - kompiluje i uruchamia program - wymienia przykłady prostych struktur danych - deklaruje zmienne liczbowe i posługuje się nimi w kodzie źródłowym	- zapisuje program w czytelnej postaci - stosuje komentarze - rozumie pojęcia implementacja, kompilacja, uruchomienie, testowanie - zna zasady poprawnego programowania - rozróżnia struktury danych deklaruje zmienne i definiuje typy złożone - rozumie znaczenie i działanie podstawowych instrukcji wybranego języka - wprowadza dane do programu i wyprowadza wyniki - definiuje i wykorzystuje funkcje i procedury - rozróżnia i poprawia błędy kompilacji i błędy wykonania - zapisuje algorytm iteracyjny w języku programowania - zapisuje algorytm z rozgałęzieniami w języku programowania	- zapisuje złożone algorytmy w języku programowania - zna rekurencyjne realizacje prostych algorytmów - rozumie i stosuje zasady programowania strukturalnego - określa zasięg działania zmiennej - przekazuje parametry do procedur i funkcji - wyprowadza dane z procedur i funkcji - wykorzystuje operatory arytmetyczne i logiczne - wykorzystuje funkcje matematyczne w obliczeniach - wykorzystuje złożone struktury danych w programach - programowo obsługuje pliki - korzysta z systemu pomocy - testuje działanie tworzonych programów	- zna różnicę między językiem wysokiego poziomu a wewnętrznym - wskazuje różnicę między kompilatorem a interpreterem - określa rolę procesora i pamięci wewnętrznej w działaniu programów - realizuje złożone algorytmy - programuje algorytmy numeryczne i sortowania - kontroluje wprowadzanie danych do programu - dobiera właściwie struktury danych	- samodzielnie rozwija umiejętności programistyczne - stosuje i łączy ze sobą złożone struktury danych - dzieli program na podprogramy - stosuje i modyfikuje znane rozwiązania w zmienionych lub nowych sytuacjach - realizuje złożone projekty programów - ocenia efektywność działania programu
------------------------	--	---	---	--	---