

Wymagania edukacyjne z przedmiotu Informatyka w zakresie rozszerzonym dla uczniów liceum i technikum (absolwentów gimnazjum)

Część 1.

Uwaga! W wymaganiach każdej z wyższych ocen zwierają się również wymagania na oceny niższe.

Temat (rozumiany jako lekcja)	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:
Zanim rozpoczniesz naukę, czyli powtórka z gimnazjum (rozdział 1)	Zajęcia nie są oceniane, służą diagnostyce poziomu wiedzy i umiejętności uczniów. Jest to niezbędne w przypadku konieczności stosowania indywidualizacji pracy z uczniem.				
Praca w zespole z wykorzystaniem sieci (rozdział 2)	- posługuje się pocztą elektroniczną i komunikatorami - wie, do czego służy usługa FTP	- korzysta z usługi FTP i dysku sieciowego - przeszukuje fora dyskusyjne	- uczestniczy w forach dyskusyjnych - umie wymienić właściwości i zastosowanie chmur informatycznych	- umie korzystać z kursów e-learningowych - wie, jak zorganizować pracę zespołu w sieci - podaje przykłady rozmaitych stron z dokładnym określeniem ich rodzaju i przeznaczenia	- zakłada ciekawe wątki na forach dyskusyjnych dotyczące informatyki i odpowiada na pytania forumowiczów
Drużyna w chmurach, czyli jak technologia wpływa na zmianę sposobu pracy (rozdział 3)	- zakłada konto w chmurze informatycznej	- umie wybrać odpowiednią chmurę informatyczną na podstawie określonych wymagań	- stawia wymagania chmurze informatycznej - korzysta z jej podstawowych programów	- organizuje pracę zespołu w chmurze informatycznej np. z wykorzystaniem kalendarza	- kieruje pracami zespołu przygotowującego i wybierającego chmurę informatyczną do konkretnego projektu
Praca w zespole z wykorzystaniem sieci czyli chmury informatyczne (rozdział 4)	- loguje się do chmury informatycznej - wie, jakie programy oferuje dana chmura	- zakłada konto w darmowej chmurze informatycznej np. w Google lub Windows Live	- wykorzystuje programy z chmury informatycznej do redagowania tekstów, rysowania itp. - korzysta z dysku w chmurze do przechowywania plików	- wykorzystuje chmurowe narzędzia do komunikowania się w zespole w celu wspólnej realizacji projektu - przenosi dokumenty z chmury do lokalnego komputera oraz edytuje dokumenty zaimportowane	- stosuje zaawansowane narzędzia edytorów z chmury informatycznej
Praca w zespole z wykorzystaniem sieci, czyli jak współużytkować pliki i dokumenty?	- edytuje tekst wspólnie z innymi członkami zespołu, jeśli taka możliwość zostanie mu udostępniona	- udostępnia pliki z dysku chmury innym użytkownikom	- udostępnia dokumenty członkom zespołu oraz nadaje im odpowiednie uprawnienia do edycji	- zarządza pracą zespołu współdzielącego dokument	- korzysta z różnych chmur informatycznych, w których przechowuje, edytuje i współdzieli dokumenty

(rozdział 5)					
Systemy operacyjne dla PC czyli charakterystyka i porównanie systemów operacyjnych Windows i Linux (rozdziały 6,7,8)	- określa podobieństwa i różnice pomiędzy interfejsami użytkownika różnych systemów operacyjnych	- posługuje się podstawowymi funkcjami systemu Linux lub Mac OS - instaluje program z Windows Live Essentials, np. Poczta systemu Windows	- uruchamia system operacyjny w wirtualnej maszynie - posługuje się monitorem zasobów systemu - instaluje wybrane, potrzebne do pracy programy za pośrednictwem Centrum oprogramowania systemu Linux - zna podstawowe cechy systemu Android	- korzysta z systemu operacyjnego uruchomionego w wirtualnej maszynie i wie, jakie to tworzy ograniczenia - zna i omawia warstwowy model systemu operacyjnego - posługuje się podstawowymi poleceniami systemowymi, takimi jak ls, mkdir, rmdir, pwd, cd, wykonywanymi za pośrednictwem Terminala systemu Linux	- modyfikuje i sprawnie dobiera parametry wirtualnej maszyny w zależności od potrzeb uruchamianego w niej systemu - korzysta z programów narzędziowych systemu Linux
Higiena systemów operacyjnych, czyli jak konserwować i konfigurować systemy. (rozdział 9 z wykorzystaniem treści z r. 6,7,8)	- omawia sposoby dbania o higienę dysku twardego - chroni komputer przed wirusami	- przywraca system, korzystając z punktu przywracania systemu - tworzy punkt przywracania systemu za pomocą narzędzia systemowego	- przeprowadza selektywną aktualizację systemu, odrzucając mniej znaczące elementy oferowane przez producenta - sprawnie posługuje się programami narzędziowymi, w tym CCleaner, do utrzymania odpowiedniego stanu systemu operacyjnego - kasuje niepotrzebne pliki - naprawia błędy w rejestrach i przywraca system od punktu przywracania	- przeprowadza defragmentację dysku komputera za pomocą programu systemowego Defragmentator dysku	- wykorzystuje programy narzędziowe do operacji na dyskach twardech i ich konserwacji np. defragmentacji, usuwania błędnych wpisów, przywracania systemu itp.
Warstwy kluczem do sukcesu, czyli jak modele organizują	- nazywa warstwowe modele sieci	- opisuje warstwy modelu OSI,	- porównuje oba modele sieci informatycznych i	- umie opisać funkcje ramki i nagłówek i	- dokładnie opisuje sposób transportu informacji w

pracę sieci? (rozdział 10)		drogę informacji w komunikacji sieciowej i warstwowy model TCP/IP	opisuje różnice ramki i nagłówków i urządzenia sieciowe w modelu TCP/IP	urządzenia sieciowe w modelu TCP/IP - umie określić adres sieci na podstawie maski	sieciach komputerowych TCP/IP
Identyfikacja komputera w sieci, czyli wiem, kim jesteś (rozdział 11)	- wie, jaką rolę pełni adres IP w sieciach komputerowych - zna podstawowe pojęcia sieciowe np. DNS, MAC	- umie skonfigurować połączenie z Internetem na podstawie znajomości maski, adresu bramy i DNS - ustawia automatyczne łączenie się komputera z wykrytą siecią i Internetem	- używa polecenia tracert i programu diagnostycznego np. VisualRoute Lite Edition do śledzenia drogi połączenia sieciowego z dowolną stroną internetową - posługuje się poleceniem ipconfig w celu odczytania pełnej konfiguracji karty sieciowej danego komputera	- posługuje się poleceniem ping do sprawdzenia połączenia sieciowego z komputerem w sieci i określenia adresu fizycznego serwera dowolnej strony www - posługuje się poleceniem ipconfig w celu odczytania pełnej konfiguracji karty sieciowej danego komputera - konfiguruje router i łączy za jego pośrednictwem sieć lokalną z Internetem	- sprawnie konfiguruje ustawienia routera sieciowego dołączonego do Internetu - tworzy sieć z zastosowaniem przełączników sieciowych
Protokoły sieciowe i zabezpieczenia sieci bezprzewodowych (rozdział 12)	- wie, do czego służą i nazywa podstawowe urządzenia sieci komputerowej, w tym także bezprzewodowe punkty dostępowe	- konfiguruje podstawowe urządzenia sieci bezprzewodowej, Access Point i karty sieciowe	- zabezpiecza sieć bezprzewodową w dostępnych standardach, w tym WPA i WPA2 z zastosowaniem PSK	- wykorzystuje program diagnostyczny np. inSSIDer do analizy ruchu w sieci bezprzewodowej i ustalenia optymalnego kanału	- zmienia kanały pracy sieci bezprzewodowej- zmienia funkcje Access Pointa na urządzenie odbierające sygnał sieci
Zasady administrowania siecią klient-serwer (rozdział 13, temat realizowany w czasie 2 lekcji)	- zna pojęcie sieć klient-serwer i wie, na czym polega praca takiej sieci	- uruchamia konsolę MMC Windows, zna funkcje szablonów zabezpieczeń	- posługuje się konsolą MMC systemu Windows, tworząc szablony zabezpieczeń i odpowiednio blokując konta - dodaje przystawki zwiększające możliwości MMC	- chroni konta przez wielokrotnymi próbami wpisywania haseł	- poznał i prezentuje na lekcji zaawansowane opcje konsoli MMC np. kreuje nowe zadania

<p>Podstawowe usługi sieciowe – udostępnianie zasobów i protokoły przesyłania plików (rozdział 14 i 15)</p>	<p>- wie, jakie elementy i zasoby można udostępniać innym użytkownikom sieci</p>	<p>- udostępnia w sieci foldery z plikami za pomocą opcji udostępniania systemu Windows, odróżnia HTTP od HTML</p>	<p>- wie, czym jest serwer IIS - włącza w systemie internetowe usługi informacyjne - odczytuje dane udostępnione przez innego użytkownika w ramach działania serwera IIS</p>	<p>- udostępnia dane w sieci za pośrednictwem serwera IIS</p>	<p>- udostępnia strony www z komputera za pomocą IIS</p>
<p>Rozwój elektronicznych środków przesyłania informacji (rozdział 16)</p>	<p>- zna podstawowe formaty przesyłanych informacji w sieciach komputerowych - uruchamia odbiór internetowych stacji radiowych</p>	<p>- wie, na czym polega transmisja strumieniowa i jakie zastosowanie znajduje RSS - wie, jakimi formatami dźwięku posługują się internetowe stacje radiowe</p>	<p>- wie, jak zabezpieczane są pliki PDF z e-książkami - odbiera nagłówki RSS za pomocą e-mail i czytników RSS np. Paseczek - odbiera transmisje strumieniowe w sieci</p>	<p>- wykorzystuje transmisje strumieniowe do nadawania audio lub wideo w sieci za pośrednictwem transmisji strumieniowej</p>	<p>- wykorzystuje różne programy do emisji strumieniowych - wykorzystuje darmowe serwery transmisji strumieniowych dla amatorskich rozgłośni radiowych</p>
<p>Cyfrowe barwy (rozdział 17)</p>	<p>- odróżnia wśród różnych kart karty graficzne, zwracając uwagę na rodzaj gniazd, i nazywa je - wie, że podstawowymi kolorami używanymi do wyświetlania obrazu są RGB</p>	<p>- zna addytywny model kolorów RGB i umie określić na podstawie opcji edytorów zawartość składowych w danym kolorze</p>	<p>- wie, jaka jest różnica między addytywnym a subtraktywnym modelem barw</p>	<p>- wie, jaki wpływ na ilość otrzymanych kolorów ma ilość bitów używanych do kodowania pojedynczej barwy modelu - zna podstawy i zastosowanie modelu percepcyjnego np. CIE LUV w edytorach grafiki</p>	<p>- określa warunki do uzyskania wiernego oryginałowi wydruku grafiki barwnej</p>
<p>Mapy i matematyka w grafice komputerowej, czyli różne metody jej tworzenia (rozdział 18)</p>	<p>- wie czym są piksele i subpiksele i jakie mają zastosowanie w wyświetlaniu obrazu na ekranie monitora</p>	<p>- wie, czym jest raster bitmapy - określa zastosowanie skanera i wpływ parametrów skanowania na jakość otrzymanego obrazu - wykazuje na przykładach różnice pomiędzy grafiką rastrową a wektorową</p>	<p>- omawia na przykładach wpływ parametrów grafiki, w tym rozdzielczości i ilości kolorów, na jej jakość oraz ograniczenia jej zastosowania - omawia zastosowanie edytorów grafiki rastrowej i wektorowej z uwzględnieniem cech obu</p>	<p>- dobiera właściwie rozdzielczość grafiki do różnych zastosowań, w tym także obrazów skanowanych, i uzasadnia te decyzje- odróżnia pojęcia rozdzielczości rozumianej jako rozmiar grafiki a liczonej w ilości pikseli na cal - wymienia zalety i wady</p>	<p>- uzasadnia różne wielkości plików dla grafiki rastrowej i wektorowej</p>

			formatów	grafiki rastrowej i wektorowej w zależności od okoliczności ich zastosowania	
Formaty plików graficznych, czyli grafika w różnych postaciach (rozdział 19)	<ul style="list-style-type: none"> - wymienia podstawowe, najczęściej używane formaty plików graficznych z uwzględnieniem rodzaju zapisanej grafiki - wie, które z nich są charakterystyczne dla fotografii 	<ul style="list-style-type: none"> - wymienia podstawowe formaty plików graficznych charakterystycznych dla najpopularniejszych edytorów grafiki - wie, czym jest RAW - zna podstawowe cechy popularnych formatów plików graficznych 	<ul style="list-style-type: none"> - wykazuje wpływ stopnia kompresji pliku graficznego na jakość grafiki i wielkość pliku - porównuje cechy popularnych formatów plików graficznych - konwertuje pliki graficzne do innych formatów z pomocą prostych programów 	<ul style="list-style-type: none"> - ustala kompromis pomiędzy jakością, stopniem kompresji a jakością grafiki, a także wielkością pliku - wie, w jakim przypadku należy użyć formatu bezstratnego 	<ul style="list-style-type: none"> - świadomie stosuje wszystkie opcje programu Easy Graphic Converter do zmiany parametrów pliku graficznego - zna inne programy o podobnych funkcjach
Narzędzia do edycji grafiki rastrowej. Kolory na zdjęciach. (rozdziały 20 i 21)	<ul style="list-style-type: none"> - umie wymienić kilka programów do edycji grafiki rastrowej - wie, czym są warstwy 	<ul style="list-style-type: none"> - porównuje narzędzia różnych edytorów grafiki rastrowej, wskazując analogie i różnice - poprawnie nazywa podstawowe narzędzia edytorów - wie, do czego służą histogramy i krzywe kolorów 	<ul style="list-style-type: none"> - umie uruchomić edytor grafiki rastrowej w chmurze informatycznej np. pixlr.com - wskazuje podobieństwa darmowych i komercyjnych edytorów grafiki - odczytuje histogramy zdjęć, określając poziom ich naświetlenia 	<ul style="list-style-type: none"> - wskazuje zalety i wady darmowych edytorów i określa zakres ich zastosowania - pokazuje na przykładzie kolorowania zdjęcia zastosowanie i działanie warstw - omawia na przykładzie zastosowanie i działanie filtrów edytorów grafiki rastrowej - analizuje histogramy dla zdjęć – stosuje krzywe kolorów do korekty lub zmiany barw na fotografiach 	<ul style="list-style-type: none"> - podaje więcej przykładów zastosowania warstw i omawia ich znaczenie dla tworzenia grafiki komputerowej - stosuje histogramy i krzywe kolorów w różnych wersjach do korekty barwnej zdjęć- wie, czym jest HDR i jak korzystać z tej techniki
Retusz o korygowanie zdjęć, czyli poprawianie rzeczywistości. (rozdziały 22 i 23)	<ul style="list-style-type: none"> - uruchamia i wczytuje zdjęcia do edytora Picasa i GIMP - odnajduje opisane w podręczniku narzędzia i używa ich, nie osiągając zadowalających rezultatów 	<ul style="list-style-type: none"> - zmienia jasność, kontrast i inne parametry zdjęcia za pomocą automatycznych narzędzi prostego edytora np. Picasa - retuszuje zdjęcie, nie 	<ul style="list-style-type: none"> - ustawia prawidłowy kontrast i jasność zdjęcia cyfrowego za pomocą suwaków narzędzia edytora grafiki np. GIMP - stosuje narzędzie zaznaczanie do retuszu 	<ul style="list-style-type: none"> - koryguje jasność i kontrast zdjęcia za pomocą krzywych - ustawia prawidłowo parametry narzędzi zaznaczania i używa ich do retuszu fotografii 	<ul style="list-style-type: none"> - używa innych poza wycinaniem narzędzi do retuszowania fotografii - retuszuje zdjęcia na bardzo dobrym poziomie - umie retuszować zdjęcia za pomocą różnych

		osiągając zadowolających rezultatów	fotografii - koryguje geometrię zdjęcia z zadowolającym wynikiem - używa narzędzi edytora, np. pieczętki, do usuwania niedoskonałości zdjęcia lub drobnych detali - retusz fotografii posiada widoczne błędy	- korzysta przy retuszu z warstw - zmienia parametry, np. kolory pojedynczych elementów zdjęcia, np. zmienia kolor oczu - świadomie stosuje zmianę parametrów wykorzystywanych narzędzi, np. zaokrąglenia i przezroczystości, w celu poprawy jakości retuszu - dobrze retuszuje zdjęcie	programów
Efekty artystyczne na zdjęciach, czyli jak używać efektów i filtrów w edycji fotografii. (rozdział 24)	- wie, jak znaleźć w różnych edytorach zbiory filtrów - wie, jaką rolę pełnią w edycji grafiki	- stosuje filtry edytorów bez korekty ich właściwości	- stosuje filtry edytorów z regulacją ich parametrów w celu uzyskania zadowolających efektów	- stosuje filtry różnych edytorów, w tym także z chmury informatycznej - świadomie stosuje filtry w celu uzyskania zaplanowanego efektu	- wykorzystuje filtry w ramach warstw, uzyskując zaplanowane efekty
Wykorzystanie edytora grafiki rastrowej do tworzenia elementów graficznych – baner (rozdział 25 przewidziany do realizacji w czasie 2 lekcji)	- używa narzędzia wstawiania napisów w edytorze grafiki rastrowej - zapisuje napis do pliku w formacie graficznym	- przekształca napis za pomocą narzędzi edytora np. przez zaznaczanie i usuwanie - doбира odpowiednie tło i jego kolorystykę	- umieszcza napis na określonym tle - do kompozycji wykorzystuje warstwy	- używa filtru Mapa obrazu do wygenerowania fragmentu kodu strony internetowej wywołującego baner w przeglądarce	- na jednym banerze tworzy kilka pól aktywnych dla różnych odnośników
Wykorzystanie edytora grafiki rastrowej do tworzenia elementów graficznych. Ozdobny napis. Techniki łączenie elementów graficznych. (Rozdziały 26, 27 i 28)	- tworzy napis ozdobny wyłącznie za pomocą pojedynczo używanych filtrów - umie przenieść zaznaczony fragment do innego rysunku	- tworzy wypukły napis zawierający wady, np. mało naturalny efekt wypukłości - umie skalować przeniesiony do innego rysunku fragment grafiki umieszczony na osobnej warstwie	- tworzy ozdobny napis z wykorzystaniem filtrów i warstw - precyzyjnie zaznacza obiekty przeznaczone do przeniesienia do innego rysunku lub zdjęcia - używa jako tła elementów wyciętych z innego zdjęcia lub rysunku; efekt połączenia jest dobry, widać drobne wady kompozycji	- tworzy atrakcyjny napis zawierający naturalnie wyglądające wypukłości - tworzy wklęsły napis umieszczony na tle w formie nagłówka strony lub dokumentu - łączy elementy z różnych rysunków lub zdjęć w jedną kompozycję - używa gradientów jako tła - używa różnych	- tworzy znak wodny z zastosowaniem warstwy z kanałem alfa

				edytorów, w tym GIMP i pixlr.com.; efekt końcowy jest atrakcyjny wizualnie	
Narzędzia do edycji grafiki wektorowej (rozdział 29)	- zna podstawowe narzędzia edytorów grafiki wektorowej - rozumie istotę edytowania wektorowego i wyświetlania rysunku na ekranie	- porównuje narzędzia różnych edytorów grafiki wektorowej, znajduje analogie i różnice	- opisuje różne rodzaje edytorów grafiki wektorowej i ich zastosowania, w tym edytor OpenOffice.org Draw, LibreOffice Draw i CadStd Lite, SketchUp InkScape	- klasyfikuje edytory ze względu na obszar zastosowania, w tym grafika użytkowa, projektowanie i rysunek techniczny, edytory 3D	- podaje przykłady komercyjnych programów do edycji grafiki wektorowej, w tym programów do projektowania dla techników
Projekt plakatu promującego szkołę z wykorzystaniem edytora grafiki wektorowej (rozdziały 30, 31 przewidziane do realizacji w czasie 3 lekcji)	- sporządza projekt plakatu i realizuje go prostymi metodami; efekt nie jest zadowalający	- tworzy prosty logotyp szkoły z zastosowaniem narzędzia kreślenia figur geometrycznych	- tworzy logotypy szkoły, np. tarczę, używając krzywych (przekształcania odcinków) - umieszcza obiekty na tle - eksportuje plakat do postaci mapy bitowej; efekt końcowy jest estetyczny, zawiera drobne niedoskonałości	- używa filtrów programu Inkscape do przekształcania obiektów, np. napisów - prawidłowo rozmieszcza elementy na plakacie - tworzy dobrze skomponowany, pozbawiony niedoskonałości plakat	- przygotowuje plakat do wydruku
Wektorowa grafika trójwymiarowa (rozdział 32 przewidziany do realizacji w czasie 2 lekcji)	- wie, na czym polega różnica między edytorami 2D a 3D - zna przeznaczenie programu ketchup i umie odczytać tutoriale producenta	- korzystając z opcji Pomocy i podręcznika, tworzy podstawowe figury geometryczne – prostopadłościan, walec, kula - przemieszcza je na ekranie w odpowiedniej płaszczyźnie - eksportuje grafikę 3D do postaci bitmapy	- tworzy elementy nawigacyjne prezentacji lub stron internetowych za pomocą edytora grafiki 3D, np. strzałki z napisami	- tworzy modele 3D budynków - używa tekstur do wypełniania płaszczyzn - zachowuje proporcje i skalę	- modeluje rysunki przedstawiające rzeczywiste budynki z okolicy z zachowaniem proporcji
Konwertowanie plików filmowych (rozdział 33)	- zna podstawowe formaty zapisu plików wideo - wie, czym jest kodek i jakie ma znaczenie dla jakości i odtwarzania filmu	- używa programu Any Video Converter do zmiany formatu pliku z zastosowaniem domyślnych parametrów	- zmienia rozdzielczość klatek w filmie wideo z zastosowaniem AVC	- konwertuje plik wideo do różnych formatów, z uwzględnieniem zmiany rozdzielczości klatek, stopnia kompresji i kodeka	- konwertuje pliki wideo do formatów urządzeń mobilnych, np. odtwarzaczy mp4, telefonów itp.

Arkusz kalkulacyjny – analiza danych	<ul style="list-style-type: none"> - tworzy proste arkusze składające się z tabel - zna podstawowe formuły 	<ul style="list-style-type: none"> - tworzy tabelę do wpisywania wartości funkcji np. kwadratowej z uwzględnieniem automatycznego wypełniania danymi za pomocą myszki 	<ul style="list-style-type: none"> - tworzy wykres funkcji na podstawie danych wpisanych do tabeli - prawidłowo opisuje osie wykresu - bada wpływ zmiany argumentów na kształt funkcji 	<ul style="list-style-type: none"> - bada wpływ zmiany argumentów i parametrów funkcji na jej kształt i miejsce w układzie współrzędnych 	<ul style="list-style-type: none"> - tworzy i bada wykresy innych funkcji np. trygonometrycznych
Analiza danych w Excel (rozdział 34 przewidziany do realizacji w czasie 2 lekcji)	<ul style="list-style-type: none"> - zna podstawowe rodzaje i kształty wykresów w arkuszu - wie, jakie jest ich zastosowanie 	<ul style="list-style-type: none"> - umie dostosować rodzaj wykresu do prezentowanych danych i zastosować go bez zmiany parametrów 	<ul style="list-style-type: none"> - importuje dane do arkusza ze stron internetowych - korzysta z arkuszy w chmurach informatycznych - zmienia parametry zastosowanego wykresu 	<ul style="list-style-type: none"> - dostosowuje wygląd wykresu do dokumentu, w którym ma być zastosowany - poprzez zmianę parametrów zwiększa czytelność i użyteczność wykresu - eksportuje wykresy do innych dokumentów 	<ul style="list-style-type: none"> - zmienia wygląd wykresu, stosując odpowiednio dobrane tekstury - eksportuje dane i wykresy do arkuszy w chmurach
Wizualizacja wyników doświadczeń, czyli jak arkusz pomaga zrozumieć zjawiska (rozdział 35 i 36)	<ul style="list-style-type: none"> - gromadzi wyniki doświadczeń w przygotowanej tabeli 	<ul style="list-style-type: none"> - przygotowuje odpowiednio sformatowaną tabelę do wpisania wyników doświadczeń na podstawie jego opisu 	<ul style="list-style-type: none"> - opracowuje formuły zgodne ze wzorami dotyczącymi doświadczeń - odpowiednio formatuje pola tabeli z wynikami - wizualizuje wyniki doświadczeń za pomocą wykresów 	<ul style="list-style-type: none"> - formatuje wykresy dla zwiększenia ich czytelności - postępuje zgodnie z zasadami tworzenia wykresów dotyczących doświadczeń: poprawnie opisuje osie, podaje jednostki, operuje kolorem 	<ul style="list-style-type: none"> - sprawdza i ocenia poprawność otrzymanych wyników z prawami fizyki itp. - ocenia jakość wykonania ćwiczenia, błędy w pomiarach, na podstawie wykresów
Szyfrowanie danych (rozdział 37)	<ul style="list-style-type: none"> - wie, na czym polega szyfrowanie danych - umie odczytać zakodowany tekst na podstawie opisanej zasady dekodowania i klucza 	<ul style="list-style-type: none"> - rozumie pojęcie certyfikatu publicznego - opisuje jego zastosowanie - zna pojęcie klucz publiczny 	<ul style="list-style-type: none"> - opisuje różnice pomiędzy kluczem publicznym symetrycznym a asymetrycznym i zna schemat ich użycia - zabezpiecza pliki tekstowe w czasie ich zachowywania na dysku 	<ul style="list-style-type: none"> - wie, czym jest podpis elektroniczny i w jakich okolicznościach się go stosuje - definiuje certyfikat klucza publicznego 	<ul style="list-style-type: none"> - opisuje sposób zdobycia podpisu elektronicznego i przypadki, w których może on być stosowany
Naturalny Kod Binarny i kod liczbowy U2	<ul style="list-style-type: none"> - wie, na czym polega zasada budowania kodu w 	<ul style="list-style-type: none"> - zna wagi bitów w bajcie i na tej podstawie 	<ul style="list-style-type: none"> - zna sposób zamiany liczby dziesiętnej na NKB 	<ul style="list-style-type: none"> - konwertuje liczby z zakresu większego od 	<ul style="list-style-type: none"> - opisuje znaczenie kodów binarnych dla rozwoju

(rozdział 38)	oparciu o wagi na przykładzie kodu dziesiętnego i naturalnego kodu binarnego - korzysta z kalkulatora dla programisty w celu dokonywania obliczeń i konwersji w różnych kodach liczbowych	określa odpowiadającą danej liczbę dziesiętną - wie, jaka jest zależność informatycznej jednostki pojemności bajt, kilobajt, megabajt itd. i relacja w stosunku do rzeczywistej wartości wielokrotności liczby, czyli wie, że 1kB nie jest równy 1000B - zna podstawy tworzenia kodu U2	za pomocą dzielenia przez podstawę kodu - określa postać liczby NKB na podstawie dziesiętnej i odwrotnie, korzystając z właściwości liczb, znajomości wag bitów - zamienia ujemne liczby dziesiętne na kod U2	jednego bajta - operuje na liczbach binarnych dodatnich i ujemnych	techniki cyfrowej - wskazuje analogie dotyczące budowy NKB i maszyn cyfrowych
Kody binarne w praktyce informatyka (rozdział 39)	- umie dodawać liczby binarne z zakresu 4-bitowego - zna różnicę pomiędzy dodawaniem arytmetycznym a sumą logiczną dwóch liczb binarnych	- dodaje liczby binarne w zakresie bajtowym - wykonuje sumę i iloczyn logiczny liczb binarnych	- wykonuje mnożenie liczb binarnych; zna i rozumie prawa algebry Boole'a dotyczące podstawowych działań logicznych	- wykonuje działania arytmetyczne i logiczne na liczbach binarnych z dowolnego zakresu - wyjaśnia zastosowanie działań logicznych na liczbach binarnych z punktu widzenia programisty	- wykonuje działania arytmetyczne na liczbach binarnych z uwzględnieniem znaku liczby
Szesnastkowy system liczbowy (rozdział 40)	- wie, jakie zastosowanie ma w informatyce kod szesnastkowy - wie, jak powstaje kod szesnastkowy w oparciu o wagi pozycji	- wie, jakie są zależności pomiędzy NKB a kodem szesnastkowym - konwertuje NKB na szesnastkowy i odwrotnie	- ilustruje zastosowanie kodu szesnastkowego w praktyce informatyka na podstawie adresów sieciowych - używa polecenia IPCONFIG	- wskazuje inne zastosowania kodu szesnastkowego w związku z niewielką ilością znaków potrzebnych do zapisu liczby HEX, np. kolory w edytorze, adresy w urządzeniach sieciowych i oprogramowaniu serwerów	- wykonuje proste działania arytmetyczne w kodzie szesnastkowym
Litery i ułamki w postaci binarnej, czyli kod ASCII i konwersja kodów (rozdział 41)	- zna znaczenie kodu ASCII w praktyce informatyka	- umie użyć kombinacji klawiszy do wprowadzenia do edytowanego dokumentu znaku za pomocą kodu ASCII	- zamienia ułamki dziesiętne na postać binarną - stosuje zapis liczby za pomocą mantysy i uzasadnia takie	- wyznacza błąd, jakim obarczona jest konwersja postaci liczb binarnych i dziesiętnych - wie, jaka jest różnica pomiędzy zapisem	- wyjaśnia okoliczności, w jakich stosuje się zapis liczb zmiennoprzecinkowych - wymienia ich wady i zalety

			rozwiązanie	zmiennie- a stałoprzecinkowym dla liczb binarnych ułamkowych	
Elektronik, informatyka, komunikacja, czyli wpływ technologii na rozwój cywilizacyjny społeczeństw (rozdział 42)	- zna najważniejsze fakty z historii komputerów i cyfrowych środków łączności	- operuje nazwami urządzeń będących ważnymi osiągnięciami w historii komputerów, np. Altair, Spectrum itp.	- opisuje proces integracji systemów łączności z globalną siecią Internet na podstawie telefonii, telewizji i radia - opisuje znaczenie rozwoju chmur informatycznych	- opisuje możliwe do realizacji w przyszłości pomysły dotyczące interfejsu użytkownika, sposobu komunikowania się z urządzeniami elektronicznymi - określa obszary zastosowania chmur informatycznych	
E-learning – platforma zdalnego nauczania Moodle (rozdział 43)	- rozumie istotę e-learningu - umie korzystać z platformy e-learningowej z uprawnieniami studenta - zakłada konto użytkownika i korzysta z przygotowanych przez nauczyciela materiałów	- przesyła do platformy pliki z konta Student - umie konfigurować swój profil, zmieniać hasło dostępu itp.	- buduje strukturę kursu e-learningowego, zakłada kategorie kursów i konfiguruje kursy - dodaje użytkowników o uprawnieniach studenta	- umieszcza w strukturze kursów pliki z materiałami do nauki, tworzy quizy z najczęściej stosowanymi formatami, np. wielokrotnego wyboru, prawda fałsz itp.	- opracowuje quiz sprawdzający wiedzę z przedmiotu informatyka lub dowolnego innego według wskazówek nauczyciela - przeprowadza sprawdzian z jego wykorzystaniem
Prawo do prywatności i przestępczość elektroniczna, czyli zagrożenia wynikające z dostępu do Internetu i rozwoju informatyki (Rozdział 44)	- wie, jak postępować, by nie naruszać prawa do prywatności drugiej osoby - stosuje te zasady w Internecie i w realnym świecie - dba o swoje dane w sieci; nie podaje bez konieczności swojego adresu, numeru telefonu i innych danych - wie, kiedy i w jakich okolicznościach może być narażony na działalność cyberprzestępców - zna podstawowe zasady ochrony przed	- zna nazwy aktów prawnych zapewniających prawo do prywatności i ochrony wizerunku - wie, jakie inne wartości są chronione, np. zawartość korespondencji, w tym elektronicznej, dane osobowe - zna podstawowe metody działania cyberprzestępców i umie się przed nimi bronić (Phishing, wyłudzenie)	- odnajduje w sieci akty prawne zapewniające prawo do prywatności i wskazuje w nich paragrafy tego dotyczące - nazywa i charakteryzuje najczęściej spotykane formy przestępstw internetowych - stosuje dostępne na rynku zabezpieczenia przed działalnością przestępczą	- wie, jakich fotografii i innych materiałów zawierających wizerunek można używać w publikacjach - wskazuje strony agencji, w których można legalnie nabyć zdjęcia do własnych publikacji - wie, których ze zdjęć z własnego archiwum można użyć bez naruszania prawa do prywatności - wie, jak reagować na zauważone w sieci ślady przestępstw, np.	- zna najnowsze treści i podstawowe zapisy nowych umów międzynarodowych i uregulowań związanych z ochroną prywatności; referuje je przed klasą - interesuje się zagadnieniami ochrony przed przestępstwami w sieci - dzieli się swoją wiedzą z kolegami

	<p>przestępcami w sieci – ostrożność, dobre układanie haseł itp.</p> <p>- nie rozpowszechnia dzieł chronionych prawem autorskim i własności bez zezwolenia</p> <p>- wie, gdzie szukać pomocy w razie stania się ofiarą cyberprzestępcy(800 100 100, helpline.org.pl)</p>			Cyberbullying, przestępstwa seksualne itp.	
Zagrożenia wynikające z rozwoju informatyki (rozdział 45)	<p>- wie, na czym polegają uzależnienia od różnych form działalności w sieci Internet</p> <p>- wie, do kogo zwrócić się o pomoc w przypadku podejrzenia siebie o uzależnienie od sieci i komputerów</p>	<p>- zna mechanizmy prowadzące do rozwoju uzależnień sieciowych</p> <p>- wskazuje gry, które mogą potencjalnie prowadzić do uzależnienia, i wskazuje ich cechy, które o tym decydują</p>	<p>- wymienia inne zagrożenia, w tym hazard, nadmierne uczestnictwo w czatach i portalach społecznościowych, uzależnienie od nowych informacji, zakupy sieciowe, licytacje, uzależnienie od środków komunikacji – komórek, sms itp.</p>	<p>- wie, jakie zagrożenia mogą się zdarzyć w trakcie korzystania z bankowości elektronicznej</p> <p>- świadomie diagnozuje swoje postępowanie i stosunek do zagrożeń</p>	<p>- zauważa objawy uzależnień i stara się im przeciwdziałać</p> <p>- odnajduje w sieci strony organizacji pomagających wyjść z nałogów sieciowych</p>
Perspektywy pracy i kształcenia w zawodzie informatyka (rozdział 46)	Umie określić swoje umiejętności i braki w dziedzinie informatyki. Planuje rozwój swoich zainteresowań. Wie, jaką pracę może podjąć wykształcony informatyk w przemyśle i innych działach gospodarki. Znajduje w sieci propozycje pracy dla informatyków i analizuje warunki przyjęcia i wymagania stawiane przez pracodawców.				

Wymagania edukacyjne cz.2

Temat (rozumiany jako lekcja)	Wymagania konieczne (ocena dopuszczająca)	Wymagania podstawowe (ocena dostateczna)	Wymagania rozszerzające (ocena dobra)	Wymagania dopełniające (ocena bardzo dobra)	Wymagania wykraczające (ocena celująca)
I. Podstawy programowania					
Edytor i kompilator, Dev-C++	Uczeń:	Uczeń:	Uczeń: – sprawnie korzysta z	Uczeń: – samodzielnie i	Uczeń: – zna i stosuje inne

<p>(1) Edytor programu Dev-C++ (2)</p>	<p>– definiuje kod źródłowy oraz pojęcie kompilatora – korzystając z pliku pomocy i podręcznika, odnajduje najczęściej używane opcje dotyczące edycji programu; – wie, czym jest kompilacja;</p>	<p>– odnajduje i zna zastosowanie podstawowych opcji edytora Dev-C++; – edytuje tekst programu komputerowego w edytorze Dev-C++;</p>	<p>możliwości edytora Dev-C++ takich jak Szukaj, Sprawdź składnie, Zamień; – omawia istotę kodu źródłowego i jego kompilacji; – stosuje zasady zapisu programu w edytorze Dev-C++;, stosując podział na linie i wcięcia w odpowiednich miejscach;</p>	<p>sprawnie korzysta ze wszystkich opcji edytora Dev-C++; – omawia funkcje wszystkich opcji środowiska programistycznego Dev-C++;</p>	<p>edytory dla programistów;</p>
<p>Tworzenie kodu źródłowego i budowa programu (3)</p>	<p>– zna strukturę programu w C++ z podziałem na bloki; – wie, na czym polega kompilacja programu;</p>	<p>– wskazuje w kodzie programu poszczególne części i opisuje ich funkcje; – wyjaśnia istotę blokowej struktury programu;</p>	<p>– wyjaśnia znaczenie i funkcje poszczególnych części programu komputerowego w C++; – samodzielnie przeprowadza kompilację i uruchomienie programu komputerowego w języku C++; – korzysta z niektórych skrótów klawiszowych Dev-C++;</p>	<p>– kompiluje program, korzystając z pracy krokowej; – biegle korzysta ze skrótów klawiszowych środowiska Dev-C++;</p>	<p>– omawia różnice pomiędzy programowaniem strukturalnym a obiektowym; – w pełni korzysta z innych środowisk programistycznych podczas tworzenia kodu i kompilacji programu;</p>
<p>Identyfikatory w C++ (4)</p>	<p>– umie zdefiniować pojęcie identyfikatora; – zna jego znaczenie dla budowy programu; – definiuje nazwy zmiennych z pomocą nauczyciela;</p>	<p>– poprawnie definiuje nazwy identyfikatorów, używając właściwych znaków; – poprawnie używa identyfikatorów;</p>	<p>– poprawnie interpretuje komunikaty o błędach w definicjach nazw zmiennych, wykorzystując system pomocy i literaturę;</p>	<p>– właściwie dobiera nazwy zmiennych, biorąc pod uwagę ich przeznaczenie i przyjmowane wartości; – samodzielnie poprawnie interpretuje komunikaty o błędach związane z definiowaniem nazw zmiennych;</p>	<p>– bezbłędnie definiuje nazwy zmiennych w języku C++;</p>

Typy proste (5)	<ul style="list-style-type: none"> – rozróżnia typy standardowe i definiowanymi przez programistę; – wie, czym różnią się typy proste od złożonych; – zna podstawowe typy dla liczb całkowitych i rzeczywistych; – korzystając z pomocy nauczyciela, poprawnie dobiera podstawowe typy proste i definiuje je w programie; – zna najważniejsze cechy stałych i zmiennych; 	<ul style="list-style-type: none"> – wymienia nazwy wszystkich typów prostych: całkowity, znakowy, logiczny i rzeczywisty; – definiuje typy, korzystając z tabeli z podręcznika; – zna przeznaczenie poszczególnych typów; – wie, czym różnią się stałe od zmiennych; 	<ul style="list-style-type: none"> – samodzielnie definiuje zmienne najważniejszych typów; – zmienne rzadziej używanych typów definiuje, korzystając z podręcznika; – wyświetla zawartość zmiennych i stałych; 	<ul style="list-style-type: none"> – samodzielnie podaje zakresy liczb dla poszczególnych typów; – zna zasady definiowania zmiennych wszystkich typów prostych w języku C++; – samodzielnie i trafnie dobiera nazwy i definiuje stałe; 	<ul style="list-style-type: none"> – samodzielnie i biegle definiuje wszystkie typy proste języka C++;
Zmienne i deklaracje (6)					
Stałe i definicje (7)					
Operatory i wyrażenia (8)	<ul style="list-style-type: none"> – podaje definicję wyrażenia; – wie, jaka jest różnica pomiędzy wyrażeniem całkowitym a rzeczywistym; – zna podstawowe operatory; – zapisuje wyrażenia z użyciem operatorów; – z pomocą nauczyciela poprawnie zapisuje wyrażenia z użyciem inkrementacji i dekrementacji; 	<ul style="list-style-type: none"> – poprawnie i samodzielnie zapisuje proste wyrażenia na z wykorzystaniem operatorów; – poprawnie nadaje wartości w skróconej formie; 	<ul style="list-style-type: none"> – korzystając z podręcznika lub odpowiednich tabel, zapisuje złożone wzory matematyczne za pomocą operatorów; – zna priorytety operatorów; – korzystając z opcji pomocy, analizuje komunikaty o błędach dotyczących wyrażzeń; 	<ul style="list-style-type: none"> – samodzielnie tworzy dowolne wyrażenia łączące operatory stałe, zmienne i wartości funkcji; – sprawnie posługuje się operatorami przypisania, inkrementacji i dekrementacji; – samodzielnie poprawnie odczytuje komunikaty o błędach dotyczących wyrażzeń i poprawia błędy; 	<ul style="list-style-type: none"> – samodzielnie analizuje fragmenty programów zawierające wyrażenia, szacując ich wartość dla różnych wartości zmiennych i stałych; – szacuje wynik działania operatorów logicznych; – samodzielnie i bezbłędnie zapisuje wyrażenia z użyciem wszystkich operatorów;
Operatory przypisania i inkrementacji/ dekrementacji (9)					

	– zna zasady używania skróconej formy nadawania wartości;				
Konwersja typów w wyrażeniach (10)	– zna ogólne zasady konwersji typów;	– korzystając z tabeli potrafi określić typ wyrażenia, kierując się zasadą konwersji z „mniejszego” do „większego” typu;	– kierując się zasadami konwersji określa typ wyrażen w których dane są różnego typu; – tworzy wyrażenia ze zmiennych różnego typu;	–określa typ rozbudowanych wyrażen; – posługuje się konwersją do tworzenia własnych zmiennych potrzebnych w trakcie pisania programów;	– samodzielnie bez użycia kompilatora jest w stanie określić poprawność i typ skomplikowanych wyrażen złożonych z różnych typów danych;
Instrukcje proste – zmieniamy wartość zmiennych (11)	– prawidłowo posługuje się instrukcją przypisania i zapisuje ją; – z pomocą nauczyciela układa proste programy z wprowadzaniem i wyprowadzaniem danych oraz prostymi obliczeniami;	– korzystając z podręcznika, przypisuje wartości jednej zmiennej do drugiej zmiennej; – układa proste programy bez kontroli zakresu zmiennych;	– samodzielnie i prawidłowo układa proste programy obliczające wartość wyrażenia z danymi wprowadzonymi do programu za pomocą klawiatury; – wyświetla wynik obliczeń na ekranie; – prawidłowo ocenia zakres wyniku obliczeń (z wykorzystaniem przykładów analizy z podręcznika);	– samodzielnie układa program z kontrolą zakresu zmiennych; – samodzielnie układa programy przypisujące wartości wyrażen do zmiennych; – zna zasady rzutowania;	– posługuje się rzutowaniem w trakcie pisanie programów w języku C++;
Instrukcje warunkowe – rozgałęziamy działania (12)	– opisuje działanie instrukcji warunkowej; – z pomocą nauczyciela interpretuje algorytmy zawierające bloki decyzyjne;	– korzystając z przykładów w podręczniku, układa programy z instrukcją warunkową na podstawie algorytmu z blokiem decyzyjnym;	– samodzielnie układa programy z wykorzystaniem instrukcji warunkowej na podstawie algorytmu; – korzystając z opcji pomocy i przykładów z podręcznika, układa programy z instrukcjami warunkowymi	– samodzielnie układa programy z instrukcjami warunkowymi złożonymi i zagnieżdżonymi; - wykorzystuje w programach instrukcje wielokrotnego wyboru;	– samodzielnie opracowuje proste algorytmy z blokami decyzyjnymi i na ich podstawie układa programy w języku C++;
Instrukcja wyboru – case (13)					

			złożonymi i zagnieżdżonymi; - posługuje się instrukcją wyboru;		
Instrukcje powtarzania – (pętla) powtarzamy działania w programie - while (14)	<ul style="list-style-type: none"> – omawia i rozumie różnice pomiędzy for, while oraz do...while; – podaje przykłady zastosowań for, while oraz do...while; – definiuje zmienną sterującą i omawia jej znaczenie w for; – omawia znaczenie warunku w pętlach; 	<ul style="list-style-type: none"> – umie wybrać rodzaj instrukcji powtarzania do rozwiązania określonego problemu; – pamięta o deklaracji zmiennej sterującej; – układa proste programy z pętlami, korzystając z przykładów rozwiązań podobnych problemów; 	<ul style="list-style-type: none"> – prawidłowo układa fragmenty programów z zastosowaniem funkcji w pętlach instrukcji powtarzania, np. wczytywanie znaku z klawiatury, oczekiwanie na wciśnięcie klawisza itp.; – prawidłowo wybiera instrukcję realizującą pętlę z określoną lub nieokreśloną liczbą powtórzeń; 	<ul style="list-style-type: none"> – układa programy, prawidłowo wykorzystując instrukcje powtarzania do budowy funkcjonalnego programu, – używa instrukcji powtarzania do układania funkcji i procedur wywoływanych przez program główny, np. sprawdzania, czy został wciśnięty klawisz, podano prawidłowe hasło itp.; 	<ul style="list-style-type: none"> – samodzielnie opracowuje algorytmy, które mogą być realizowane za pomocą instrukcji powtarzania, i na ich podstawie układa programy w języku C++; – przewiduje w algorytmach konieczność stosowania instrukcji for, while lub do...while;
Instrukcje powtarzania – (pętla) powtarzamy działania w programie – do ... while (15)					
Instrukcje powtarzania – (pętla) powtarzamy działania w programie – for (16)					
Operator warunkowy i przecinkowy (17)	<ul style="list-style-type: none"> - zna konstrukcję wyrażenia warunkowego i wyrażen zawierających operator przecinkowy; - przy pomocy nauczyciela określa poprawność powyższych wyrażen; 	<ul style="list-style-type: none"> - samodzielnie potrafi określić wynik prostych wyrażen zawierających operator warunkowy i przecinkowy; 	<ul style="list-style-type: none"> - potrafi określić wynik rozbudowanych wyrażen zawierających operator warunkowy i przecinkowy; - zapisuje schematy blokowe zawierające okna decyzyjne za pomocą wyrażen warunkowych; 	<ul style="list-style-type: none"> - potrafi interpretować i konstruować skomplikowane wyrażenia w użyciu nowych operatorów; 	<ul style="list-style-type: none"> - samodzielnie bez użycia kompilatora bezbłędnie posługuje się wyrażeniami zawierającymi operatory warunkowy i przecinkowy; - tworzy zagnieżdżone konstrukcje warunkowe;
Priorytet operatorów (18)	<ul style="list-style-type: none"> - rozumie pojęcie priorytetu operatorów; - potrafi określić kolejność wykonywania podstawowych działań w wyrażeniach arytmetycznych; 	<ul style="list-style-type: none"> - zna działanie operatorów języka C++; - używając tabeli z podręcznika jest w stanie określić prawidłowy wyniki wyrażen 	<ul style="list-style-type: none"> - samodzielnie określa priorytet operatorów języka C++; - prawidłowo interpretuje wyrażenia zawierające poszczególne grupy operatorów; 	<ul style="list-style-type: none"> - bezbłędnie porównuje priorytety operatorów; - określa prawostronność lub lewostronność operatorów; - sprawnie posługuje się operatorami języka 	<ul style="list-style-type: none"> - posługuje się operatorami działań binarnych; - wykorzystuje w programowaniu operatory działające na strukturach i wskaźnikach;

		arytmetycznych;	- tworzy wyrażenia zawierające operatory języka C++;	C++;	
Tablice – zaczynamy definiować swoje typy (złożone) (19)	<ul style="list-style-type: none"> – podaje definicję typu tablicowego i z pomocą nauczyciela podaje różne przykłady zastosowania tablic; – z pomocą nauczyciela analizuje definicję typu tablicowego, wskazując najważniejsze jej elementy; – obrazowo opisuje zastosowanie tablic jedno-, dwu-, trój- i więcej wymiarowych; – graficznie wyjaśnia budowę tablic; 	<ul style="list-style-type: none"> – zna znaczenie słów kluczowych w definicjach tablic; – określa wartość zmiennej w tablicy za pomocą wartości typu indeksowego; – umie zaplanować tablicę o odpowiednim do rozwiązywanego problemu rozmiarze; – samodzielnie definiuje własne typy tablicowe jednowymiarowe; 	<ul style="list-style-type: none"> – definiuje własne tablice zmiennych różnych typów i rozmiarów; – prawidłowo zastępuje zmienną tablicową osobno występującą zmienną, uzasadniając swoją decyzję i wykazując zasadność takiego postępowania; – podaje przykłady definiowania różnych tablic i objaśnia zasadność przyjęcia takiej konstrukcji zmiennej tablicowej, np. używanych w przykładzie z podręcznika; 	<ul style="list-style-type: none"> – prawidłowo zastępuje zmienną tablicową osobno występującą zmienną, uzasadniając swoją decyzję i wykazując zasadność takiego postępowania; – nadaje wartości początkowe zmiennym tablicowym; – samodzielnie modyfikuje zmienną tablicową, np. poprzez zwiększenie rozmiaru; – analizuje zajęcie pamięci przez daną tablicę; – umie wskazać błąd w deklaracjach tablic zarówno składniowy, jak i logiczny; 	<ul style="list-style-type: none"> – samodzielnie opracowuje algorytmy, które wykorzystują tablice, i na ich podstawie układa programy w języku C++; – stosuje tablice wielowymiarowe do grupowania danych, np. wprowadzanych imion, nazwisk itp.;
Tablice – definiujemy własne typy złożone (20)					
Tablice – definiujemy własne typy złożone – ćwiczenia (19) (20)					
Wskaźniki (21)	<ul style="list-style-type: none"> – zna idee budowy zmiennych wskaźnikowych; – z pomocą nauczyciela definiuje zmienną wskaźnikową; 	<ul style="list-style-type: none"> – prawidłowo deklaruje zmienną wskaźnikową; – ustawia wskaźniki na adresach zmiennych; 	<ul style="list-style-type: none"> – poprawnie definiuje wskaźniki różnych typów; – poprawnie interpretuje kody źródłowe z zastosowaniem wskaźników; 	<ul style="list-style-type: none"> – programuje z użyciem zmiennych wskaźnikowych; – rozumie problem przekazywania wartości pomiędzy wskaźnikami i niebezpieczeństwo wskazywania na ten sam adres pamięci; 	<ul style="list-style-type: none"> – samodzielnie planuje użycie zmiennych wskaźnikowych do rozwiązywania problemów informatycznych; – samodzielnie układa programy, stosując zmienną wskaźnikową;
Wskaźniki i tablice (22)	<ul style="list-style-type: none"> - zna zależność pomiędzy nazwą tablicy a wskaźnikiem; - wymienia poznane 	<ul style="list-style-type: none"> - z pomocą nauczyciela interpretuje zapisy wskaźników odwołujące się to tablic; 	<ul style="list-style-type: none"> - samodzielnie interpretuje zapisy wskaźników odwołujące się to tablic; 	<ul style="list-style-type: none"> - sprawnie posługuje się wskaźnikami do tablic; - tworzy zmienną łańcuchową i wykonuje 	<ul style="list-style-type: none"> - używa w kodach źródłowych wskaźników do tablic; - sprawnie posługuje się łańcuchami znaków;

	funkcje działające na tekstach;	- zna różnice pomiędzy wskaźnikiem a nazwą tablicy; - określa działanie poszczególnych funkcji modułu <string.h>	- tworzy zapisy z użyciem wskaźników do tablic;	na nich podstawowe operacje; - posługuje się funkcjami działającymi na tekstach;	
Funkcje – piszemy własne podprogramy (23)	– zna budowę funkcji; – wie, na czym polega wywołanie funkcji;	– wywołuje funkcje w treści programu; – analizuje działanie poszczególnych funkcji;	– stosuje unikalne nazwy zmiennych wewnątrz bloku; – prawidłowo układa mało skomplikowane funkcje wykorzystujące globalne i lokalne zmienne; – wywołuje funkcje z parametrami; – prawidłowo i czytelnie dobiera nazwy zmiennych w deklaracjach funkcji; – wie, jak działa przekazywanie przez wartość; – wie, jak działa przekazywanie przez wskaźnik;	– omawia istotę deklaracji wewnętrznych zmiennych i stałych, typów i podprogramów lokalnych; – układa i wywołuje funkcje z parametrami i bez; – prawidłowo układa funkcje zwracające wartość i void wykorzystujące globalne i lokalne zmienne; – stosuje unikalne nazwy zmiennych wewnątrz bloku; – układa funkcje z przekazywaniem parametrów przez wartość lub wskaźnik;	– samodzielnie planuje użycie funkcji do rozwiązywania problemów informatycznych; – samodzielnie układa funkcje, stosując zmienne lokalne i globalne;
Funkcje –zmienne lokalne i globalne, przesłanianie zmiennych (24)	– zna różnicę pomiędzy działaniem i zastosowaniem funkcji zwracającej wartość i funkcji void; – określa przypadki, w których niezbędne jest zastosowanie funkcji zwracającej wartość i funkcji void;				
Funkcje – przekazywanie parametrów (25)					
Struktury – grupujemy dane w jedną całość (26)	– określa funkcję jaką odgrywają struktury w programach komputerowych;	– definiuje typ strukturalny i pojedynczą strukturę; – podaje przykłady typów strukturalnych;	– omawia strukturę zmiennych strukturalnych; – deklaruje struktury składające się z różnych danych; – definiuje zmienne strukturalne;	– deklaruje struktury składające się z różnych danych; – omawia i stosuje strukturę tablicową zmiennych strukturalnych; – uzasadnia stosowanie przez zmienną (* i &) w	– samodzielnie planuje użycie struktur złożonych do łączenia w grupy danych, które mają być ze sobą powiązane, i wybiera dane na podstawie analizy algorytmu;
Struktury - zmienne strukturalne (27)	– odróżnia od siebie typ strukturalny i zmienną typu strukturalnego; – z pomocą	– definiuje zmienne strukturalne; – podaje przykłady zmiennych	– podaje przykłady		

	podręcznika analizuje przykłady typów strukturalnych;	strukturalnych;	odwołania się do pola;	przypadku zmiany wartości zmiennej strukturalnej wewnątrz funkcji;	
Operacje wejścia i wyjścia – strumienie i pliki (28)	<ul style="list-style-type: none"> – omawia poszczególne etapy działań na plikach; – wie, jaka jest różnica pomiędzy plikami tekstowymi a binarnymi w C++; – zna sens sekwencyjnego dostępu do elementów w pliku; 	<ul style="list-style-type: none"> – deklaruje zmienne plikowe dla plików różnych typów; – wymienia i charakteryzuje etapy działań z plikami; - zna podstawowe funkcje do pracy z plikami; 	<ul style="list-style-type: none"> – otwiera i tworzy pliki; – używa funkcji fprintf i fscanf, fwrite i fread, – programuje pobranie, modyfikacje i zapis danych do pliku; – prawidłowo zamyka pliki; 	<ul style="list-style-type: none"> – posługuje się funkcjami: fputs, fputc, fgets i fgetc, fseek, ftell i foef; – świadomie stosuje dwa sposoby dostępu do pliku – sekwencyjny i swobodny; – prawidłowo dobiera i wykorzystuje funkcje działające na plikach; 	<ul style="list-style-type: none"> – samodzielnie planuje wykorzystanie odpowiednich funkcji działających na plikach do realizacji programów na podstawie analizy algorytmów;
Operacje wejścia i wyjścia – wykonujemy działania na plikach (29)					
Operacje wejścia i wyjścia – wykonujemy działania na plikach tekstowych (30)					
Operacje wejścia i wyjścia – wykonujemy działania na plikach (31)					
Wykrywamy błędy (debugowanie) (32)	<ul style="list-style-type: none"> – zna istotę debugowania i jego znaczenie w pracy programisty; – podaje przykłady, w których wskazane jest użycie debugera i uruchamiania krokowego programu; – wie, czym jest pułapka; 	<ul style="list-style-type: none"> – omawia różnicę pomiędzy opcjami debugowania (F7 i F8); – debuguje prosty program i wskazuje miejsca wyświetlania danych debugowania, np. wartości zmiennych; – świadomie stosuje debugowanie do wykrywania błędów w krótkim programie bez 	<ul style="list-style-type: none"> – wykorzystuje debugowanie do analizy błędów w programie z wywołaniem funkcji z niewielką ilością zmiennych; – umie kontrolować wartości zmiennych w trakcie debugowania; – uruchamia program metoda krokową, kontrolując zawartości zmiennych; 	<ul style="list-style-type: none"> – uruchamia program metoda krokową, kontrolując zawartości zmiennych; – umie uruchomić debugowanie od dowolnego miejsca programu; – debuguje lub pomija w tym procesie podprogramy; – odnajduje błędy za pomocą debugera lub pracy krokowej, 	<ul style="list-style-type: none"> – biegle korzysta z debugera, analizując tylko wybrane fragmenty programu. – typuje miejsca w programie, w których należy szukać błędów, i kontroluje je debugerem;

		funkcji;	– umie uruchomić debugowanie od dowolnego miejsca programu; – debuguje lub pomija w tym procesie podprogramy;	porównując wartości zmiennych z przewidywanymi przez algorytm; – umie kontrolować wartości zmiennych w trakcie debugowania;	
II. Dynamiczne struktury danych					
W świecie wskaźników (1)	– wie, jakie są mechanizmy odwoływania się do zmiennych dynamicznych za pomocą wskaźników; – podaje przykłady takich zastosowań typów wskaźnikowych;	– wskazuje różnice pomiędzy zmiennymi wskaźnikowymi statycznymi a dynamicznymi;	– posługuje się operatorami new i delete; – wie, jak działają powyższe operatory; – analizuje gotowe przykłady deklaracji zmiennych wskaźnikowych; – deklaruje zmienne wskaźnikowe; – umie odwołać się do zmiennej dynamicznej za pomocą zmiennej wskaźnikowej;	– umie analizować błędy powstałe podczas tworzenia programu ze zmiennymi dynamicznymi; – umie analizować błędy powstałe podczas tworzenia programu ze zmiennymi dynamicznymi; – wie, czym jest spowodowana utrata dostępu do zmiennej dynamicznej; – zna znaczenie stałej nil; – określa zastosowania zmiennych wskaźnikowych takie jak odwołanie do zmiennych dynamicznych, użycie w instrukcji przypisania, użycie w relacji, jako wynik funkcji;	– samodzielnie układa programy z zastosowaniem zmiennych dynamicznych; – podaje uzasadnienie użycia zmiennych dynamicznych we własnym rozwiązaniu problemu informatycznego;
Deklarujemy zmienne wskaźnikowe (2)					
Tworzymy pierwsze zmienne dynamiczne (3)					
Co jeszcze powinienś wiedzieć o wskaźnikach? (4)					
Dynamiczne struktury danych (5)	– umie podać przykład zastosowania dynamicznej struktury danych;	– zna znaczenie rekordu w tworzeniu struktur dynamicznych jako zmiennej dynamicznej;	– definiuje i deklaruje w programie struktury dynamiczne; – łączy zmienne dynamiczne na	– samodzielnie łączy zmienne dynamiczne; – używa zmiennej wskaźnikowej do wypełniania pola	– samodzielnie układa programy z zastosowaniem struktur dynamicznych; – podaje uzasadnienie
Dynamiczne struktury danych – zapis kodu					

(6)		<ul style="list-style-type: none"> – wie, czym jest zmienna wskaźnikowa i umie opisać jej znaczenie w dynamicznej strukturze danych; – wie, czym jest pole wskaźnikowe w rekordzie; 	<ul style="list-style-type: none"> podstawie przykładów z podręcznika; – usuwa zmienną dynamiczną; – zmienia wartość zmiennej dynamicznej; 	<ul style="list-style-type: none"> rekordu dynamicznego; – wypełnia pola rekordu dynamicznego; 	<ul style="list-style-type: none"> użycia struktur dynamicznych we własnym rozwiązaniu problemu informatycznego;
<ul style="list-style-type: none"> Stos – ostatni wchodzi, pierwszy wychodzi LIFO (7) Kolejka – pierwszy wchodzi, pierwszy wychodzi (FIFO) (8) 	<ul style="list-style-type: none"> – umie opisać istotę stosu; – umie podać różnicę pomiędzy LIFO a FIFO; 	<ul style="list-style-type: none"> – zna budowę stosu i jego zastosowanie; – opisuje rolę wskaźników i zmiennych wskaźnikowych w tworzeniu i obsłudze kolejek; 	<ul style="list-style-type: none"> – wie, jak zbudować stos dla zmiennych dynamicznych; – wie, jak przeglądać stos i jak usuwać zmienne dynamiczne; – wie, jak tworzyć w swoich programach kolejkę zmiennych; – wie; na czym polega wstawianie elementów do kolejki, jej przeglądanie i usuwanie elementów; 	<ul style="list-style-type: none"> – samodzielnie buduje stos dla zmiennych dynamicznych; – zna funkcję zmiennej wskaźnikowej w adresowaniu wierzchołka stosu; – samodzielnie tworzy program przeglądający stos; – samodzielnie tworzy program usuwający zmienne dynamiczne; – tworzy w swoich programach kolejkę zmiennych; 	<ul style="list-style-type: none"> – wie, jak zachowuje się komputer, gdy zostaje wywołane przerwanie, i wie, jaka w tym rola stosu procesora; – stosuje w swoich programach stos programowy; – identyfikuje w algorytmach i problemach informatycznych zagadnienia, w których należy zastosować FIFO lub LIFO;
<ul style="list-style-type: none"> Lista jednokierunkowa (9) Lista dwukierunkowa, a nawet cykliczna (10) 	<ul style="list-style-type: none"> – wskazuje różnicę pomiędzy listą jedno- i dwukierunkową; – zna dokładnie różnicę pomiędzy kolejką a stosem; 	<ul style="list-style-type: none"> – wymienia cechy listy; – analizuje graficzne, podręcznikowe przedstawienie działań na listach; – wymienia niektóre problemy informatyczne, w których stosuje się listy; 	<ul style="list-style-type: none"> – umie porządkować listy jednokierunkowe; – umie definiować w programie listy dwukierunkowe; – na podstawie podręcznika analizuje przykład programu z listą cykliczną; – definiuje strukturę dla listy dwukierunkowej; – wie, czym jest i w 	<ul style="list-style-type: none"> – wstawia i usuwa elementy z listy jedno- i dwukierunkowej; – z powodzeniem stosuje w programach listy w tym także cykliczne; 	<ul style="list-style-type: none"> – identyfikuje w algorytmach zagadnienia, które mogą być zrealizowane z wykorzystaniem list;

			<p>jakich przypadkach znajduje zastosowanie lista cykliczna i podaje zasadniczą cechę listy cyklicznej;</p> <ul style="list-style-type: none"> – samodzielnie analizuje kod programu, w którym zastosowano listy; 		
Drzewo (11)	<ul style="list-style-type: none"> – opisuje analogię struktury drzewiastej do struktury folderów w systemie operacyjnym; 	<ul style="list-style-type: none"> – wymienia cechy drzewa; – analizuje graficzne, podręcznikowe przedstawienie struktury drzewiastej; 	<ul style="list-style-type: none"> – wymienia niektóre problemy informatyczne, w których stosuje się drzewa; - interpretuje kod źródłowy zawierający implementacje struktur drzewiastych; 	<ul style="list-style-type: none"> - modyfikuje kody źródłowy zawierający implementacje struktur drzewiastych; 	<ul style="list-style-type: none"> - potrafi zastosować struktury drzewiaste do swoich potrzeb;
III Bazy danych					
Tabele, wiersze i klucze (1)	<ul style="list-style-type: none"> – wie, czym jest baza danych; – zna znaczenie tabel, wierszy i kluczy dla bazy danych; 	<ul style="list-style-type: none"> – wie, jakie znaczenie ma prawidłowe zaplanowanie i projektowanie tabel w kontekście poprawności i szybkości działania bazy danych; – wybiera odpowiednie nazwy dla pól tabeli; – operuje słownictwem znamionym dla baz danych takim jak rekordy, pola, klucze; 	<ul style="list-style-type: none"> – wskazuje pola, które jednoznacznie identyfikują rekordy, np. PESEL, numer telefonu itp.; – prawidłowo ustala klucze i identyfikatory dla tabel; – zna budowę rekordu tabeli; – wie, czym są klucze, klucze główne i czym się one różnią; 	<ul style="list-style-type: none"> – trafnie dobiera pola do tworzenia kluczy zgodnie z założeniami projektowymi; – zna zastosowanie pól sztucznych autonumerowanych; – buduje tabele bazy zgodnie z założeniami projektowymi; 	<ul style="list-style-type: none"> – samodzielnie projektuje rozbudowane tabele dla baz danych z jednoczesnym funkcjonalnym wyborem pól kluczy;
Projektujemy bazę danych. Pierwsza i druga postać normalna. (2)	<ul style="list-style-type: none"> – rozumie pojęcie i sens normalizacji oraz zna jej główny cel; 	<ul style="list-style-type: none"> – wskazuje różnice pomiędzy 1., 2., 3. i 4. postacią normalną; 	<ul style="list-style-type: none"> – ustala klucz główny dla 1. postaci normalnej tabeli; – umie przeprowadzić proces normalizacji do 	<ul style="list-style-type: none"> – zna zależności pomiędzy polami niekluczowanymi a kluczem głównym; – wskazuje miejsca, w 	<ul style="list-style-type: none"> – przeprowadza normalizację tabeli do postaci 4. normalnej z pominięciem normalizacji do postaci
Projektujemy bazę					

danych. Trzecia i czwarta postać normalna. (3)			2. postaci normalnej; – omawia rolę klucza głównego w 2. postaci normalnej;	których następuje redundancja danych; – przeprowadza normalizację kolejno do wszystkich postaci normalnych;	2. i 3.;
Projektujemy bazę danych. Określamy relacje między tabelami. (4)	– odróżnia relacje 1–1 od 1–n; – wie, czym są relacje;	– określa prawidłowe relacje pomiędzy tabelami, używając kluczy; – zna pojęcia klucza głównego i klucza obcego;	– umie określić integralność bazy danych na podstawie analizy tabel;	– wykazuje, że po poprawnym procesie normalizacji tabele są powiązane prawidłowymi relacjami;	– samodzielnie określa relacje pomiędzy tabelami na podstawie własnego projektu bazy;
Pierwsze chwile z bazą danych (5)	– wymienia nazwy darmowych i komercyjnych programów do tworzenia baz danych, w tym OpenOffice.org Base, Libre Office.org Base, Access; – umie uruchomić program do tworzenia relacyjnych baz danych, np. OpenOffice.org Base i zna rozmieszczenie opcji menu;	– zna i omawia funkcję obiektów głównego ekranu programu OpenOffice Org Base;	– posługuje się kreatorem bazy danych; – wie, jakie pliki powstają podczas tworzenia bazy;	– zna nazwy obiektów na ekranie głównym kreatora i omawia ich przeznaczenie;	– samodzielnie realizuje bazę danych według własnego projektu; – samodzielnie planuje tabele, relacje, układa kwerendy, organizuje wprowadzanie i wyprowadzanie danych z bazy; – samodzielnie testuje bazę;
Tworzymy tabele (6)	– odnajduje i uruchamia kreatora tabel;	– umie uruchomić i posługiwać się kreatorem tabel programu do tworzenia baz danych, np. OpenOffice.org Base;	– samodzielnie tworzy tabele z użyciem kreatora;	– samodzielnie tworzy tabele bez użycia kreatora;	
Tworzymy i modyfikujemy tabele (7)	– wie, do czego służy i co można przy jego pomocy osiągnąć; – tworzy tabelę z pomocą podręcznika i	– wykazuje małą	– prawidłowo ustala typy pól w tabelach; – modyfikuje nazwy tabel, pól i typów pól;	– ustala klucze dla tabel; – bezbłędnie stosuje zasady nadawania	

	nauczyciela;	samodzielność w tworzeniu tabel i myli pola;	– zna nazwy i przeznaczenie poszczególnych pól kreatora tabel; – wprowadza dane do tabeli; – prawidłowo ustala klucze, w tym główne;	nazw pól w tabeli; – prawidłowo i samodzielnie ustala klucze, w tym główne;	
Indeksujemy i określamy relacje (8)	– wyjaśnia pojęcie indeksu głównego;	– wyjaśnia, w jakim celu wprowadza się indeksowanie i relacje między tabelami; – wie, że klucze główne są indeksowane automatycznie;	– samodzielnie tworzy indeksy z wykorzystaniem kreatora; – samodzielnie ustala i tworzy relacje;	– planuje powiązania i indeksy zgodnie z założeniami wyszukiwania danych w bazie; – samodzielnie tworzy indeksy bez kreatora;	
Budujemy kwerendy z kreatora (9)	– wyjaśnia pojęcie kwerenda i jej znaczenie dla baz danych; – wie, czym jest SQL;	– układa kwerendy dla prostej bazy danych; – wyjaśnia, czym różni się kwerenda szczegółowa od skróconej; – omawia działanie najważniejszych poleceń języka SQL;	– używa kreatora do formułowania kwerend; – używa odpowiednich słów do wypełniania pól kreatora; – wybiera pola w kreatorze kwerend; – ustala porządek sortowania; – ustala warunki przeszukiwania; – sprawdza poprawność kwerendy; – tworzy tabele za pomocą języka SQL; – organizuje wydruk danych z kwerendy za pomocą SQL;	– świadomie używa wszystkich opcji kreatora w czasie tworzenia kwerendy; – przeprowadza testy poprawności działania kwerendy; – korzysta z aliasów; – samodzielnie tworzy kwerendę dla jednej i większej ilości tabel ze wszystkimi jej cechami bez użycia kreatora; – posługuje się edytorem SQL z programu OpenOffice.org Base; – tworzy kwerendę za pomocą SQL i edytora z programu OpenOffice.org Base;	
Budujemy kwerendy bez kreatora (10)					
Budujemy kwerendy za pomocą SQL (11)					

				– tworzy kwerendy z warunkiem za pomocą SQL;	
Kreujemy formularze (12)	– umie używać gotowych formularzy do wprowadzania danych;	– korzysta z najważniejszych opcji kreatora formularzy; – stosuje style kreatora dla formularzy; – prawidłowo określa nazwę formularza;	– umiejętnie wybiera pola formularza; – prawidłowo wybiera tryb wprowadzania danych; – prawidłowo rozmieszcza formaty; – prawidłowo wybiera tryby wprowadzania danych;	– definiuje podformularze;	
Drukujemy raporty (13)	– wie, do czego służą raporty; – wie, który kreator służy do organizowania wydruków raportów.	– dobiera odpowiedni wygląd raportu; – drukuje gotowe raporty; – wie, że raporty tworzy się na podstawie tabeli lub kwerendy.	– używa kreatora i wszystkich jego opcji do tworzenia raportów; – prawidłowo wybiera pola do raportu; – prawidłowo i zgodnie z charakterem danych nazywa pola; – grupuje dane w raporcie.	– organizuje sortowanie danych w raporcie; – odróżnia raporty statyczne od dynamicznych.	

UWAGA! Do uzyskania oceny celującej w niektórych tematach wymagane są informacje i umiejętności wykraczające poza treść podręcznika