

Wymagania edukacyjne na oceny śródroczne z informatyki w roku szkolnym 2024-2025 dla klasy 8

Podstawa programowa określa cele kształcenia, a także obowiązkowy zakres treści programowych i oczekiwanych umiejętności, które uczeń o przeciętnych uzdolnieniach powinien przyswoić na danym etapie kształcenia. Opisane w niej wymagania szczegółowe można przypisać do pięciu kategorii.

1. Analizowanie i rozwiązywanie problemów – problemy powinny być raczej proste i dotyczyć zagadnień, z którymi uczniowie spotykają się w szkole (np. na matematyce) lub na co dzień; rozwiązania mogą przyjmować postać planu działania, algorytmu lub programu (nie należy wymagać od uczniów biegłości w programowaniu w jakimkolwiek języku).
2. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi – uczniowie powinni w trakcie lekcji bez większych problemów wykonywać konkretne zadania za pomocą dostępnego oprogramowania, w tym sprawnie korzystać z menu, pasków narzędzi i pomocy programów użytkowych i narzędziowych, oraz tworzyć dokumenty i przedstawiać efekty swojej pracy np. w postaci dokumentu tekstowego lub graficznego, arkusza, prezentacji, programu, baz danych czy wydruku.
3. Zarządzanie informacjami oraz dokumentami – uczniowie powinni umieć wyszukiwać informacje, porządkować je, analizować, przedstawiać w syntetycznej formie i udostępniać, a także gromadzić i organizować pliki w sieci lokalnej lub w chmurze.
4. Przestrzeganie zasad bezpiecznej pracy z komputerem – uczniowie powinni przestrzegać regulaminu pracowni komputerowej oraz zasad korzystania z sieci lokalnej i rozległej, a także rozumieć zagrożenia związane z szybkim rozwojem technologii informacyjnej.
5. Przestrzeganie prawa i zasad współżycia – uczniowie powinni przestrzegać praw autorskich dotyczących korzystania z oprogramowania i innych utworów, a podczas korzystania z sieci i pracy w chmurze stosować się do zasad netykiety.

Ocenianie uczniów na lekcjach informatyki powinno być zgodne z założeniami szkolnego systemu oceniania. Uczniom i rodzicom powinny być znane wymagania stawiane przez nauczycieli i sposoby oceniania. Niniejszy dokument zawiera najważniejsze informacje, które można zaprezentować na początku roku szkolnego. Ważne jest, aby standardowej ocenie towarzyszył opis osiągnięć ucznia – szczegółowe uwagi dotyczące sposobu rozumowania, podejścia do zagadnienia. Trzeba pamiętać, że treści programowe są różnorodne. Obejmują zarówno operowanie elementami algorytmiki, jak i posługiwanie się narzędziami informatycznymi, czyli technologią informacyjną. Umiejętności te należy oceniać w sposób równorzędny, ponieważ zdarza się, że uczniowie, którzy świetnie radzą sobie z programami użytkowymi, mają duże trudności z rozwiązywaniem problemów w postaci algorytmicznej, i odwrotnie – uczniowie rozwiązujący trudne problemy algorytmiczne i potrafiący sprawnie programować słabo posługują się programami użytkowymi. Należy uświadamiać uczniom ich braki, ale wystawiając ocenę, przykładając większą wagę do mocnych stron.

Sprawdzając wiadomości i umiejętności uczniów, należy brać pod uwagę osiem form aktywności.

Forma aktywności	Częstość formy aktywności	Uwagi
zadania i ćwiczenia wykonywane podczas lekcji	na każdej lekcji	oceniać należy przede wszystkim zgodność efektu pracy ucznia nad zadaniami i ćwiczeniami z postawionym problemem (np. czy funkcja utworzona przez ucznia daje właściwy wynik), mniejsze znaczenie ma sposób rozwiązania
praca na lekcji	na każdej lekcji	oceniać należy sposób pracy, aktywność, przestrzeganie regulaminu pracowni
odpowiedzi ustne, udział w dyskusjach	czasami	
sprawdziany	po każdym dziale	mogą mieć formę testu
prace domowe	czasami	jeśli praca domowa wymaga użycia komputera, należy przypomnieć uczniom, że w razie potrzeby mogą skorzystać z komputera np. w bibliotece lub w pracowni komputerowej – w trakcie zajęć dodatkowych
referaty, opracowania, projekty	czasami	
przygotowanie do lekcji	w razie potrzeby	oceniać należy pomysły i materiały przygotowane do pracy na lekcji
udział w konkursach		nieobowiązkowa forma aktywności; przejście do kolejnych etapów powinno odpowiednio podwyższyć ocenę końcową

Opis wymagań ogólnych, które uczeń musi spełnić, aby uzyskać daną ocenę

Ocena celująca (6) – uczeń wykonuje samodzielnie i bezbłędnie wszystkie zadania z lekcji oraz dostarczone przez nauczyciela trudniejsze zadania dodatkowe; jest aktywny i pracuje systematycznie; posiada wiadomości i umiejętności wykraczające poza te, które są wymienione w planie wynikowym; w konkursach informatycznych przechodzi poza etap szkolny; w razie potrzeby pomaga nauczycielowi (np. przygotowuje potrzebne na lekcję materiały pomocnicze, pomaga kolegom w pracy); pomaga nauczycielom innych przedmiotów w wykorzystaniu komputera na ich lekcjach.

Ocena bardzo dobra (5) – uczeń wykonuje samodzielnie i bezbłędnie wszystkie zadania z lekcji; jest aktywny i pracuje systematycznie; posiada wiadomości i umiejętności wymienione w planie wynikowym; w razie potrzeby pomaga nauczycielowi (pomaga kolegom w pracy).

Ocena dobra (4) – uczeń wykonuje samodzielnie i niemal bezbłędnie łatwiejsze oraz niektóre trudniejsze zadania z lekcji; pracuje systematycznie i wykazuje postępy; posiada wiadomości i umiejętności wymienione w planie wynikowym.

Ocena dostateczna (3) – uczeń wykonuje łatwe zadania z lekcji, czasem z niewielką pomocą, przeważnie je kończy; stara się pracować systematycznie i wykazuje postępy; posiada większą część wiadomości i umiejętności wymienionych w planie wynikowym.

Ocena dopuszczająca (2) – uczeń czasami wykonuje łatwe zadania z lekcji, niektórych zadań nie kończy; posiada tylko część wiadomości i umiejętności wymienionych w planie wynikowym, jednak brak systematyczności nie przekreśla możliwości uzyskania przez niego podstawowej wiedzy informatycznej oraz odpowiednich umiejętności w toku dalszej nauki.

Uwagi dodatkowe

- Jeśli przyjęte w szkole zasady na to pozwalają, nie trzeba wymagać od uczniów prowadzenia zeszytu (należy wówczas poprosić o gromadzenie wydruków oraz notatek wykonywanych podczas lekcji w teczce lub segregatorze). Konieczne jest natomiast systematyczne zapisywanie wykonanych w pracowni ćwiczeń w określonym miejscu w sieci lokalnej lub w chmurze. Można też poprosić uczniów o przynoszenie na lekcje pamięci USB – w celu prowadzenia własnego archiwum plików.
- Warto zachęcać uczniów do samodzielnego oceniania swojej pracy – powinni umieć stwierdzić, czy ich rozwiązanie jest poprawne. W miarę możliwości należy uzasadniać oceny i dyskutować je z uczniami.
- Aby poprawić ocenę, uczeń powinien wykonać powtórnie najgorzej ocenione zadania (lub zadania podobnego typu) w trakcie prowadzonych w pracowni zajęć dodatkowych albo w domu, jeśli jest taka możliwość i można wierzyć, że dziecko będzie pracować samodzielnie.
- Uczeń powinien mieć możliwość zgłoszenia nieprzygotowania dwa razy w semestrze. Nieprzygotowanie powinno zostać zgłoszone przed rozpoczęciem lekcji (np. podczas sprawdzania obecności). Nie zwalnia ono ucznia z udziału w lekcji – jeśli to konieczne, uczniowi powinni podczas zajęć pomagać nauczyciel i koledzy.
- Uczeń, który był dłużej nieobecny, powinien w miarę możliwości nadrobić istotne ćwiczenia i zadania wykonane na opuszczonych lekcjach. Można określić, że jeśli np. liczba niewykonanych ćwiczeń przekroczy 20% wszystkich prac z danego działu, uczeń powinien to nadrobić.

Nr lekcji	Temat lekcji	Omawiane zagadnienia	Ocena	Zgodnie z wymaganiami programowymi uczniów:
1. Lekcje z HTML-em				
1.1	Jak to zrobić w HTML-u i CSS?	Programy do tworzenia stron WWW, wprowadzenie w historię języka znaczników hipertekstu (HTML) oraz kaskadowych arkuszy stylów (CSS), ogólna struktura dokumentu HTML, definiowanie stylów w dokumencie HTML (rodzaje arkuszy stylów, podstawowe znaczniki)	2	▪ z pomocą nauczyciela tworzy prosty dokument HTML.
			3	▪ wprowadza w edytorze tekstu ustawienia dotyczące kodowania znaków; ▪ samodzielnie tworzy prosty dokument HTML.
			4	▪ poprawnie stosuje elementy CSS.
			5	▪ tworzy dokument HTML zgodnie z zaleceniami W3C; ▪ wyjaśnia specyfikę różnych rodzajów kaskadowych arkuszy stylów.
			6	▪ przygotowuje prezentację wyjaśniającą rolę, jaką w historii języka HTML odegrali Tim Berners-Lee, Robert Cailliau, Håkon Wium Lie i Bert Bos, oraz cel powołania W3C.
1.2	Strona w dobrym stylu	Tworzenie dokumentu HTML z zastosowaniem CSS – definiowanie właściwości czcionki i akapitu, definiowanie jednostek miar i kolorów, osadzanie elementów graficznych, korzystanie ze znaków specjalnych	2	▪ stosuje style wpisane w celu sformatowania tekstu.
			3	▪ stosuje znaki specjalne (zwłaszcza <code>&nbsp;</code>);
			4	▪ stosuje różne jednostki miary; ▪ definiuje kolory różnych elementów dokumentu; ▪ osadza w dokumencie elementy graficzne.
			5	▪ definiuje właściwości czcionek (krój czcionki, styl czcionki, wariant czcionki, wysokość czcionki, odstępy między literami, zmiana wielkości znaków); ▪ definiuje właściwości akapitu (odstępy między wyrazami, dekorowanie tekstu, wyrównanie tekstu w poziomie).
			6	▪ wykorzystuje style wpisane, osadzone i zewnętrzne; ▪ stosuje wybór przez klasę.

Nr lekcji	Temat lekcji	Omawiane zagadnienia	Ocena	Zgodnie z wymaganiami programowymi uczniów:
1.3	Strona interaktywna	Wprowadzenie do dynamicznego HTML, tworzenie elementów interaktywnych za pomocą CSS i JavaScript, budowanie galerii z wykorzystaniem elementów interaktywnych	2	<ul style="list-style-type: none"> ▪ wyjaśnia pojęcie „dynamiczny HTML”. ▪ z pomocą nauczyciela tworzy i umieszcza na stronach HTML elementy interaktywne w CSS z wykorzystaniem pseudoklasy <code>:hover</code>.
			3	<ul style="list-style-type: none"> ▪ samodzielnie tworzy i umieszcza na stronach HTML interaktywne elementy w CSS z wykorzystaniem pseudoklasy <code>:hover</code>.
			4	<ul style="list-style-type: none"> ▪ z pomocą nauczyciela tworzy i umieszcza na stronach HTML elementy interaktywne w JavaScript z wykorzystaniem zdarzeń <code>onclick</code>, <code>onmouseover</code>, <code>onmouseout</code>.
			5	<ul style="list-style-type: none"> ▪ samodzielnie tworzy i umieszcza na stronach HTML elementy interaktywne w JavaScript z wykorzystaniem zdarzeń <code>onclick</code>, <code>onmouseover</code>, <code>onmouseout</code>; ▪ samodzielnie tworzy interaktywną galerię fotografii.
			6	<ul style="list-style-type: none"> ▪ stosuje inne dynamiczne pseudoklasy CSS; ▪ tworząc elementy interaktywne, stosuje własne rozwiązania.
1.4	Witryna WWW	Rodzaje witryn WWW, porządkowanie kodu dokumentu HTML, tworzenie witryny przez połączenie poszczególnych dokumentów HTML systemem odnośników	2	<ul style="list-style-type: none"> ▪ opisuje budowę adresu strony WWW; ▪ wyjaśnia znaczenie rozszerzenia domeny.
			3	<ul style="list-style-type: none"> ▪ wyjaśnia znaczenie nazwy <code>index.htm</code>; ▪ tworzy odnośniki tekstowe i graficzne do innych dokumentów.
			4	<ul style="list-style-type: none"> ▪ omawia strukturalną budowę dokumentu HTML; ▪ opisuje rolę znaczników: <code>header</code>, <code>nav</code>, <code>article</code>, <code>section</code>, <code>aside</code>, <code>footer</code>. ▪ z pomocą nauczyciela stosuje ww. znaczniki do tworzenia dokumentu HTML.
			5	<ul style="list-style-type: none"> ▪ samodzielnie stosuje ww. znaczniki do tworzenia poprawnej struktury dokumentu.
			6	<ul style="list-style-type: none"> ▪ tworząc witrynę WWW, pracuje samodzielnie i stosuje własne rozwiązania; ▪ kopiuje pliki składowe na serwer WWW i weryfikuje poprawność działania witryny.
1.5	Prawo w internecie	Prawo autorskie a ochrona wizerunku oraz twórczości (ochrona elementów serwisów i całych serwisów WWW, ochrona oprogramowania), wolne oprogramowanie, bezpieczeństwo w sieci	2	<ul style="list-style-type: none"> ▪ wyjaśnia konieczność chronienia utworów (np. programów, zdjęć, stron WWW).
			3	<ul style="list-style-type: none"> ▪ wyjaśnia, na czym polega naruszenie praw autorskich i jak go uniknąć.
			4	<ul style="list-style-type: none"> ▪ wyjaśnia pojęcia „dozwolony użytek prywatny” i „ochrona wizerunku”.
			5	<ul style="list-style-type: none"> ▪ wyjaśnia, czym są wolne oprogramowanie i cztery rodzaje wolności.
6	<ul style="list-style-type: none"> ▪ wyjaśnia praktyczne znaczenie najważniejszych punktów <i>Ustawy o prawie autorskim i prawach pokrewnych</i>. 			

Nr lekcji	Temat lekcji	Omawiane zagadnienia	Ocena	Zgodnie z wymaganiami programowymi uczniów:
2. Lekcje programowania				
2.1	Pisz i powtarzaj	Instalowanie programu Python, stosowanie polecenia <code>print</code> i pętli <code>for</code>	2	▪ pisze i uruchamia prosty program wypisywania tekstu na ekranie (polecenie <code>print</code>).
			3	▪ stosuje pętlę <code>for</code> .
			4	▪ wyjaśnia, jak działa funkcja <code>range</code> w zależności od liczby parametrów.
			5	▪ rysuje szlaczki i figury, wykorzystując pętlę <code>for</code> , polecenie <code>print</code> .
			6	
2.2	Programuj obliczenia	Operacje matematyczne, typy zmiennych, definiowanie funkcji bez parametru	2	▪ opisuje i odpowiednio wykorzystuje operacje matematyczne.
			3	▪ opisuje i odpowiednio wykorzystuje zmienne.
			4	▪ definiuje proste funkcje bez parametru.
			5	▪ rozwiązuje problemy z wykorzystaniem funkcji bez parametru.
			6	
2.3	Sumuj liczby	Operowanie zmiennymi, definiowanie funkcji z parametrem, stosowanie instrukcji warunkowej i podstawowych algorytmów na liczbach naturalnych	2	▪ zmienia wartość zmiennej.
			3	▪ omawia działanie parametru w funkcji.
			4	▪ definiuje funkcję z parametrem służącą do wyodrębnienia cyfr danej liczby czterocyfrowej i obliczenia ich sumy.
			5	▪ definiuje funkcję z parametrem służącą do wyodrębnienia cyfr dowolnej liczby całkowitej i obliczenia ich sumy; ▪ opisuje działanie instrukcji warunkowej i wykorzystuje ją do zbadania podzielności liczb.
			6	▪ samodzielnie rozwiązuje dodatkowe zadania programistyczne.
2.4	Liczby nie tylko doskonałe	Wykorzystywanie funkcji do obliczeń, moduł <code>math</code>	2	▪ rozumie problem znajdowania podzielników właściwych liczby.
			3	▪ korzysta z modułu <code>math</code> .
			4	▪ z pomocą nauczyciela definiuje funkcję obliczania sumy dzielników właściwych liczby podanej jako parametr.
			5	▪ definiuje funkcję wypisywania liczb doskonałych; ▪ testuje działanie funkcji dla różnych parametrów.
			6	▪ samodzielnie rozwiązuje dodatkowe zadania programistyczne.

Nr lekcji	Temat lekcji	Omawiane zagadnienia	Ocena	Zgodnie z wymaganiami programowymi uczniów:
2.5	Szukaj z Pythonem	Wyszukiwanie elementu w zbiorze uporządkowanym i nieuporządkowanym, moduł <code>random</code> , stosowanie pętli <code>while</code>	2	<ul style="list-style-type: none"> ▪ rozumie zasady gry <i>Zgadnij liczbę</i>; ▪ biorąc udział w grze, potrafi zastosować optymalną strategię.
			3	<ul style="list-style-type: none"> ▪ znajduje maksymalną liczbę kroków odgadywania danej liczby.
			4	<ul style="list-style-type: none"> ▪ losuje liczby całkowite z danego zakresu; ▪ wykorzystuje pętlę <code>while</code> do znajdowania sumy cyfr liczby.
			5	<ul style="list-style-type: none"> ▪ analizuje schemat blokowy algorytmu obliczania sumy cyfr dowolnej liczby; ▪ samodzielnie implementuje grę <i>Zgadnij liczbę</i> w Pythonie, korzystając ze wskazówek w podręczniku.
			6	<ul style="list-style-type: none"> ▪ samodzielnie rozwiązuje dodatkowe zadania programistyczne.
2.6	Zrób porządek	Porządkowanie elementów zbioru przez prosty wybór i zliczanie, wykorzystywanie <code>list</code>	2	<ul style="list-style-type: none"> ▪ opisuje porządkowanie zbioru przez proste wybieranie i zliczanie.
			3	<ul style="list-style-type: none"> ▪ opisuje, czym jest lista, i potrafi z niej korzystać.
			4	<ul style="list-style-type: none"> ▪ korzysta z funkcji związanych z listami.
			5	<ul style="list-style-type: none"> ▪ definiuje funkcje zliczania.
			6	<ul style="list-style-type: none"> ▪ samodzielnie rozwiązuje dodatkowe zadania programistyczne.