

# WYMAGANIA EDUKACYJNE

Rok szkolny 2024/2025

<b>ZAWÓD:</b>	Technik informatyk
<b>NR PROGRAMU:</b>	351203/T-5/ZSEE/Ti-5/2019.09.01
<b>PROGRAM REALIZUJE</b>	Karol Kołodziejczyk

	Kod	Nazwa	godz. /tydz.		godz. /rok
			I	II	
<b>MODUŁ:</b>	<b>M8</b>	<b>Programowanie w języku Python</b>	7	7	210

## 1. Sposoby sprawdzania osiągnięć uczniów:

### Uczeń otrzymuje oceny za (nauczyciel nie musi wykorzystywać wszystkich form oceniania):

- Wybrane prace wykonywane na zajęciach (bez zapowiedzi, która praca będzie oceniana). Ocena może być wystawiona za jedną pracę lub za kilka prac łącznie
- Odpowiedzi ustne (może być stosowany system np. jedna ocena za dwie/trzy odpowiedzi – należy to ustalić z klasą)
- Kartkówki (może być stosowany system np. jedna ocena za dwie/trzy kartkówki – należy to ustalić z klasą)
- Sprawdziany, prace klasowe (w formie testów/ prac praktycznych)
- Aktywność
- Dodatkowe prace wykonywane samodzielnie (np. w formie projektów podsumowujących)
- Udział i osiągnięcia w konkursach

## 2. Uzupelnienie szkolnych kryteriów oceniania

1. Sprawdziany/ prace klasowe są obowiązkowe.
2. Uczeń, który nie wykonywał prac ocenianych na zajęciach (nieobecności, przyczyny losowe), powinien postępować zgodnie z procedurami poprawkowymi, czyli zgłosić się do nauczyciela w celu uzupełnienia/zaliczenia pracy. Zaliczenie może być np. w formie pracy domowej, odpowiedzi ustnej lub wykonania pracy praktycznej w wyznaczonym przez nauczyciela terminie z zastrzeżeniem, że nie każde ćwiczenie musi być poprawione (decyduje nauczyciel). Po minięciu ustalonych terminów nauczyciel wstawia ocenę niedostateczną.
3. Uczeń, który otrzymał ocenę niedostateczną ze sprawdzianu/pracy klasowej ma możliwość jej poprawy w terminie ustalonym z nauczycielem w ciągu dwóch tygodni od otrzymania oceny.
4. Zadania poprawkowe powinny być analogiczne do zadań poprawianych (ale niekoniecznie takie same) i mogą być trudniejsze, ponieważ:
  - Uczeń ma więcej czasu na przygotowanie się, niż grupa wykonująca pracę w terminie podstawowym,
  - Ma możliwość dowiedzieć się, jakiego typu zadania wykonywała klasa.
5. Osoby posiadające opinie z poradni psychologiczno-pedagogicznej będą miały dostosowane wymagania z modułu w zależności od wskazań osoby wystawiającej zaświadczenie (np. wydłużenie czasu pisania sprawdzianu lub pracy klasowej).

Inne zasady oceniania: zgodnie z wewnątrzszkolnym systemem oceniania (dot. m.in. zasad i terminów oceniania, informowania o wymaganiach i zasadach oceniania, terminach i zakresie poprawek).

Osoby posiadające opinie z poradni psychologiczno-pedagogicznej będą mieli dostosowane wymagania z modułu w zależności od wskazań osoby wystawiającej zaświadczenie (np. wydłużenie czasu pisania sprawdzianu lub pracy klasowej).

Wymagania z poziomu wyższego zawierają wymagania z poziomu niższego, ocena dopuszczająca jest informacją dla ucznia, że sukces na dalszym etapie kształcenia jest możliwy pod warunkiem uzupełnienia braków.

### 3. Wymagania na poszczególne oceny

Ocena	Wymagane umiejętności	
dopuszczająca	<b>Wymagania konieczne</b>	<i>Stosowanie nabytych umiejętności na poziomie samodzielnego (odtwórczego) rozwiązywania zadań, ograniczonych do mniej zaawansowanych zagadnień.</i>
	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Uruchamia skrypty Python w terminalu</li> <li>• Poprawnie tworzy pliki python.</li> <li>• Deklaruje i wczytuje zmienne</li> <li>• Wymienia typy wbudowane (bool, int, float, str, tuple, list, set, dict)</li> <li>• Poprawnie dokonuje rzutowania typów adekwatnie do wymogów programu</li> <li>• Wymienia i poprawnie używa operatorów arytmetycznych typów liczbowych</li> <li>• Wymienia i poprawnie używa operatorów sekwencji</li> <li>• Wymienia i poprawnie używa operatorów logicznych</li> <li>• Poprawnie używa operatorów rozpakowania</li> <li>• Opisuje i poprawnie używa slice'ów</li> <li>• Opisuje i poprawnie używa range'ów</li> <li>• Poprawnie używa funkcji len, zwłaszcza w kontekście pętli for</li> <li>• Opisuje i poprawnie używa list i słowników</li> <li>• Jest w stanie sprawdzić typ za pomocą type</li> <li>• Używa funkcji print do wyświetlania wartości</li> <li>• Podaje wiele argumentów pozycyjnych printa</li> <li>• Wymienia i poprawnie stosuje argumenty nazwane printa - sep i end</li> <li>• Opisuje działanie funkcji input wraz ze zwracanym typem</li> <li>• Poprawnie używa funkcji input</li> <li>• Definiuje funkcje za pomocą słowa kluczowego def</li> <li>• Poprawnie używa return</li> <li>• Rozumie różnice między zwracaniem a wyświetlaniem</li> <li>• Poprawnie używa wyrażenia warunkowego if</li> <li>• Poprawnie używa pętli for iterując po listach/krotkach itp.</li> <li>• Poprawnie używa pętli for iterując po listach/krotkach itp.</li> <li>• Poprawnie używa pętli while</li> <li>• Tworzy własne klasy</li> <li>• Poprawnie definiuje konstruktor</li> <li>• Używa klas wbudowanych, importowanych i własnych do tworzenia instancji</li> <li>• Instaluje moduły narzędziem pip</li> <li>• Pamięta o nienazywaniu plików i folderów takimi samymi nazwami jak moduły</li> <li>• Używa polecenia import do wczytywania własnych skryptów i modułów instalowanych</li> <li>• Zna różnicę między import ... as ... a from ... import ... itp.</li> <li>• Potrafi wykrywać podstawowe błędy składniowe i logiczne.</li> <li>• Rozumie znaczenie wyjątków i potrafi odczytać podstawowe komunikaty błędów.</li> </ul>	

	<b>Wymagania podstawowe</b>	<i>Stosowanie nabytych umiejętności na poziomie samodzielnego (odtwórczego) rozwiązywania zadań, ograniczonych do mniej zaawansowanych zagadnień.</i>
	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Potrafi używać konstrukcji `try-except` do obsługi wyjątków.</li> <li>• Rozumie znaczenie `finally` i `else` w kontekście obsługi wyjątków.</li> <li>• Potrafi używać lintera Flake8 (lub bardziej złożonego) do ewaluacji kodu i dostosowuje kod do jego wymagań</li> <li>• Potrafi używać formatera Blue (lub innego, bardziej złożonego)</li> <li>• Potrafi korzystać z podstawowych modułów standardowych, np. `math`, `random`, `time`.</li> <li>• Potrafi otwierać pliki za pomocą konstrukcji with i funkcji open</li> <li>• Stosuje odpowiednie flagi otwarcia pliku w zależności od potrzeb</li> <li>• Potrafi wczytuje dane z plików txt</li> <li>• Potrafi zapisać dane do plików txt</li> <li>• Potrafi tworzyć dokumentację kodu w plikach markdown</li> <li>• Potrafi dokumentować napisane funkcje/klasy za pomocą docstringów</li> <li>• Potrafi dokumentować napisane funkcje/klasy za pomocą typehintów</li> <li>• Zna różnicę między `type(x) == y` i `isinstance(x,y)`</li> <li>• Zna i potrafi używać metod krotki</li> <li>• Zna i potrafi używać metod listy</li> <li>• Poprawnie zapisuje argumenty domyślne adekwatnie do potrzeb</li> </ul>	
<b>Dostateczna</b>	<b>Wymagania rozszerzone</b>	<i>Stosowanie nabytych umiejętności na poziomie samodzielnego rozwiązywania zadań/ problemów projektowych, konstrukcyjnych i obliczeniowych.</i>
	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Potrafi omówić klasy abstrakcyjne collection, iterator, sequence</li> <li>• Potrafi stwierdzić subklasą jakiej abc jest klasa dowolnego kontenera wbudowanego</li> <li>• Potrafi zapisywać / wczytywać pliki JSON modułem JSON</li> <li>• Potrafi zapisywać / wczytywać pliki Pickle modułem Pickle</li> <li>• Poprawnie zapisuje interpolację za pomocą f-stringów</li> <li>• Poprawnie używa funkcji enumerate</li> <li>• Tworzy środowiska wirtualne</li> <li>• Tworzy pliki zależnościowe</li> <li>• Instaluje zależności z plików zależnościowych</li> <li>• Stosuje tablice prawdy</li> <li>• Zna i potrafi używać metod stringa</li> <li>• Potrafi tworzyć generatory z definicji funkcji</li> <li>• Potrafi poprawnie zapisywać argsy i kwargsy</li> </ul>	
<b>dobra</b>	<b>Wymagania dopełniające</b>	<i>Stosowanie nabytych umiejętności na poziomie samodzielnego rozwiązywania zadań/ problemów projektowych, konstrukcyjnych i obliczeniowych.</i>
	<p>Uczeń:</p> <ul style="list-style-type: none"> <li>• Potrafi definiować i rzucać własne wyjątki.</li> <li>• Zna i potrafi używać metod zbiorów</li> <li>• Zna i potrafi używać metod słowników</li> <li>• Zna i potrafi używać operatorów zbiorów</li> <li>• Zna i potrafi używać wyrażenia trójskładnikowego</li> <li>• Zna i potrafi używać wyrażenia generatora</li> <li>• Zna i potrafi używać wyrażenia listy, zbioru i słownika</li> <li>• Zna i potrafi używać map, filtrów i redukcji</li> <li>• Zna i potrafi używać słów kluczowych break i continue</li> <li>• Potrafi używać else po pętli</li> <li>• Potrafi pisać proste testy kodu za pomocą frameworków doctest, pytest i unittest</li> <li>• Używa modułu logging do logowania</li> </ul>	
<b>bardzo dobra</b>		