

## Podstawy programowania – wymagania edukacyjne

### Ocena dopuszczająca (2)

Uczeń:

- zna podstawowe pojęcia: program, instrukcja, zmienna, stała, typ danych, kompilator, interpreter;
- rozróżnia podstawowe typy danych (liczby całkowite, zmiennoprzecinkowe, znakowe, logiczne);
- wie, czym są dane wejściowe i wyjściowe programu;
- zna pojęcie algorytmu i potrafi go opisać słownie;
- rozumie podstawowe instrukcje sterujące (sekwencja, warunek, pętla);
- zna zasady nazewnictwa zmiennych;
- rozumie pojęcie składni i semantyki języka programowania;
- zna zasady bezpieczeństwa pracy w środowisku programistycznym.

### Ocena dostateczna (3):

Uczeń spełnia wymagania na ocenę dopuszczającą oraz:

- wyjaśnia różnice między typami danych, ich zastosowanie i zakres wartości;
- opisuje działanie instrukcji warunkowych (if, if-else, instrukcje wielokrotnego wyboru);
- rozumie zasadę działania pętli (for, while, do-while) i różnice między nimi;
- zna podstawowe operatory: arytmetyczne, relacyjne, logiczne, przypisania;
- rozumie priorytet i łączność operatorów;
- wyjaśnia pojęcie tablicy jednowymiarowej i jej zastosowanie;
- rozumie podstawowe pojęcia związane z funkcjami (definicja, wywołanie, prototyp);
- zna pojęcie komentarza i jego znaczenie w kodzie.

### Ocena dobra (4):

Uczeń spełnia wymagania na ocenę dostateczną oraz:

- wyjaśnia zagnieźdżanie instrukcji sterujących i ich wpływ na przepływ programu;

- opisuje tablice wielowymiarowe i ich reprezentację w pamięci;
- rozumie mechanizm przekazywania parametrów do funkcji (przez wartość i przez referencję);
- wyjaśnia pojęcia: parametr formalny, parametr aktualny, wartość zwracana;
- zna pojęcie wskaźnika, adresu pamięci i operatorów wskaźnikowych;
- rozumie zakres widoczności zmiennych (lokalne, globalne);
- opisuje czas życia zmiennych;
- wyjaśnia zasady walidacji danych wejściowych;
- rozumie konwersje typów (jawne i niejawne).

#### **Ocena bardzo dobra (5):**

Uczeń spełnia wymagania na ocenę dobrą oraz:

- wyjaśnia związek między tablicami a wskaźnikami;
- opisuje arytmetykę wskaźników;
- rozumie mechanizm rekurencji, jej zalety i ograniczenia;
- wyjaśnia dynamiczną alokację pamięci i zarządzanie pamięcią;
- opisuje przeciążanie funkcji i domyślne wartości parametrów;
- rozumie zasady programowania strukturalnego i modularnego;
- wyjaśnia pojęcie złożoności obliczeniowej;
- analizuje organizację pamięci programu (stos);
- rozumie mechanizmy obsługi błędów;
- opisuje różnice między błędami składniowymi, logicznymi i wykonania.

#### **Ocena celująca (6):**

Uczeń spełnia wymagania na ocenę bardzo dobrą oraz:

- opisuje zaawansowane struktury danych (listy, stosy, kolejki, drzewa);
- wyjaśnia wskaźniki do funkcji i ich zastosowania;
- rozumie mechanizmy optymalizacji kodu;

- analizuje paradygmaty programowania (strukturalne, proceduralne, obiektowe);
- opisuje zasady programowania defensywnego i czystego kodu;
- wyjaśnia mechanizmy debugowania i profilowania;
- rozumie zasady tworzenia i wykorzystania bibliotek;
- posiada wiedzę o procesie kompilacji i konsolidacji;
- samodzielnie pogłębia wiedzę teoretyczną wykraczającą poza program nauczania;

#### UWAGI:

- Przedmiot ma charakter teoretyczny- nacisk położony jest na zrozumienie koncepcji i mechanizmów programowania
- Praktyczna implementacja poznanych zagadnień realizowana jest na odrębnym przedmiocie
- Ocena opiera się na znajomości teorii, umiejętności wyjaśniania pojęć i analizy problemów programistycznych