Zajęcia zgrupowaliśmy w pięć semestralnych dedykowanych modułów. Dodatkowo, w ramach modułów nie dotyczących bezpośrednio programowania (np. Systemów Operacyjnych, Systemów Baz Danych, Projektowania Witryn Internetowych) dbamy o integrację z potrzebami studentów ukierunkowanych na programowanie (np. pokazując jak przygotowywać dedykowane serwery aplikacyjne, serwery baz danych, czy też jak „rozmawiać” z serwisami REST).

Zajęcia zawsze są dostosowane do potrzeb studentów, i w grupach nie przekraczających 20 osób prowadzone są wielopoziomowo (uwzględniając również fakt, że część grup studentów może być ukierunkowana na inne technologie IT, np. na grafikę komputerową). W czasie zajęć część prezentacyjna stanowi ~ 1/3 całości, a pozostała część polega na wspieraniu indywidualnych postępów uczestników.

Oferujemy w pełni skonfigurowane środowiska programistyczne, również z możliwością dostępu zdalnego („z domu”). Kod „napisany” w trakcie zajęć jest [dostępny publicznie](https://github.com/teamWSIZ/python1-2021).
Wszystkich zainteresowanych studiami z zakresu programowania zapraszamy do zapoznania się z naszą ofertą edukacyjną oraz procedurą rekrutacji [na studia inżynierskie](https://www.wsi.edu.pl/studia-inzynierskie/) oraz [na studia podyplomowe](https://www.wsi.edu.pl/studia-podyplomowe/).

**Semestr 1 : Podstawy**

* podstawy składni i konstrukcja prostych, krótkich programów
* integracja z wydajnym środowiskiem programistycznym [pycharm](https://www.jetbrains.com/pycharm/)

 ( Link - <https://www.jetbrains.com/pycharm/> )

* podstawy ciekawych i bardzo użytecznych aplikacji (praca z csv, danymi z internetu [(requests)](https://www.youtube.com/watch?v=Pa23kKiGHH8) itp) (Link - <https://m.youtube.com/watch?v=Pa23kKiGHH8> )
* korzystanie biblioteki standardowej Pythona, jak i bibliotek dołączanych przez pip

**Semestr 2 : Wydajna praca**

* efektywne wykorzystanie elementów biblioteki standardowej
* struktury hashowane, czyli dict i set, i ich wykorzystanie w pisaniu naprawdę szybkich aplikacji
* spora liczba prostych i trudniejszych zadań pozwalających na nabranie wprawy w języku (są to też typowe zadania używane na rozmowach rekrutacyjnych),
* projektowanie interfejsów graficznych ([kivy](https://www.youtube.com/watch?v=YDp73WjNISc)), i prostych frameworków backendowych ([flask](https://www.youtube.com/watch?v=9sG0xjGwIMM)).

(Link kivy - <https://m.youtube.com/watch?v=YDp73WjNISc> )

(Link flask - <https://m.youtube.com/watch?v=9sG0xjGwIMM> )

**Semestr 3 : Obiektowo-zorientowana architektura aplikacji**

* best-practices podziału aplikacji na funkcjonalne moduły
* hierarchia obiektowa (implementacja interfejsów, rozszerzanie klas istniejących)
* obsługa wyjątków
* budowanie testów aplikacji
* [przykłady](https://github.com/teamWSIZ/oop-2021) (Link - <https://github.com/teamWSIZ/oop-2021> )

**Semestr 4 : Wydajna integracja z systemami baz danych**

* tworzenie/modyfikacja i praca z systemami baz danych w SQL (w dialekcie postgresql)
* praca z typowym systemem baz danych ([sklep dostępny na w3schools](http://www.w3schools.com/sql/)), konstrukcja zapytań SQL (Link - <https://www.w3schools.com/sql/> )
* zapytania złożone typu JOIN, klucze obce
* wersjonowanie struktury baz [alembic](https://alembic.sqlalchemy.org/en/latest/) (Link - <https://alembic.sqlalchemy.org/en/latest/> )
* wydajny framework backendowy [aiohttp](https://docs.aiohttp.org/en/stable/) współpracujący asynchronicznie z bazą danych [asyncpg](https://github.com/MagicStack/asyncpg).

 (Link aiohttp – <https://docs.aiohttp.org/en/stable/> )

 (Link asyncpg – <https://github.com/MagicStack/asyncpg> )

* testy wydajnościowe [locust](https://www.youtube.com/watch?v=SOu6hgklQRA) (Link - <https://m.youtube.com/watch?v=SOu6hgklQRA> )
* best-practices tworzenia wielo-klienckiego dostępu do danych; zachowanie integralności, transakcyjność

**Semestr 6 : Analiza danych i sztuczna inteligencja**

* kurs na podstwie biblioteki [pytorch](https://www.youtube.com/watch?v=7bU_4QBubGA)

 (Link - <https://m.youtube.com/watch?v=7bU_4QBubGA> )

* analiza i klasyfikacja obrazów

 (prawie jak tu - <https://m.youtube.com/watch?v=zRnSmw1i_DQ> )

* przewidywanie sekwencji czasowych „[predykcja przyszłości](https://www.youtube.com/watch?v=Rh7tLUzfEwA)„

 (Link- <https://m.youtube.com/watch?v=Rh7tLUzfEwA> )

* szukanie ukrytych zależności w ogromnych zbiorach danych [UMAP](https://pair-code.github.io/understanding-umap/)

( Link - <https://pair-code.github.io/understanding-umap/> )

* [przykłady](https://github.com/teamWSIZ/AI2022) (Link- <https://github.com/teamWSIZ/AI2022> )