

Чернівецький національний університет імені Юрія Федьковича
Навчально-науковий інститут фізико-технічних та комп'ютерних наук
Кафедра математичних проблем управління і кібернетики

СИЛАБУС
навчальної дисципліни

«Програмування мовою Python»
(вибіркова)

Освітньо-професійна програма «Інформаційні системи та технології»

Спеціальність 126 – Інформаційні системи та технології

Галузь знань 12 – Інформаційні технології

Рівень вищої освіти перший бакалаврський

Мова навчання українська

Розробник: д.ф.-м.н., професор кафедри математичних проблем управління і кібернетики Малик Ігор Володимирович

Профайл викладача: <https://mpuik.vercel.app/about/staff/malyk-ihor-volodymyrovych>

Контактний тел. +38(0372)509-340

E-mail: i.malyk@chnu.edu.ua

Сторінка курсу в Moodle <https://moodle.chnu.edu.ua/course/view.php?id=6045>

Сторінка курсу в Google Classroom
<https://classroom.google.com/c/NDU3Nzg3NzEyNTg1?cjc=rbt3tgx>

Консультації очні та онлайн - згідно з графіком
(за попередньою домовленістю зі студентами).

1. Анотація дисципліни (призначення навчальної дисципліни).

Курс “Програмування мовою Python” є вибірковою для студентів спеціальності 126 Інформаційні системи та технології. Метою курсу є надання студентам теоретичних основ та формування практичних навичок програмування мовою Python задач різного рівня на ЕОМ, вивчення основних положень мови програмування Python, придбання студентами знань і навиків в області розробки алгоритмів, створення, трансляції та налагодження прикладних програм, застосування бібліотек та модулів Python.

Велика увага приділяється практичному застосуванню мови для вирішення актуальних задач, на створення зрозумілого для читання коду, заохочення студентів спостерігати за новинами та подіями у спільноті розробників мови Python. Знання цієї мови програмування допоможе студентам у вивченні майбутніх дисциплін та при подальшому працевлаштуванні.

2. Мета навчальної дисципліни:

- ознайомлення студентів із основними поняттями мови програмування Python ;
- формування у студентів практичних навичок програмної реалізації розроблених алгоритмів та навичок роботи з відомими бібліотеками та модулями Python;
- формування компетентностей з використання основних положень мови програмування Python при розв’язуванні задач, які виникають при розробці програмного забезпечення та інформаційних систем.

3. Пререквізити. Для успішного засвоєння дисципліни студенти повинні попередньо на належному рівні опанувати дисципліни “Алгоритмізація та програмування”, “Об’єктно-зорієнтоване програмування”, “Бази даних”.

4. Результати навчання

Згідно з вимогами освітньо-професійної програми підготовки бакалаврів спеціальності 126 Інформаційні системи та технології студенти після вивчення навчальної дисципліни “Програмування мовою Python” повинні набути таких компетентностей:

Загальні компетентності:

- КЗ 2. Здатність застосовувати знання у практичних ситуаціях.
- КЗ 5. Здатність вчитися і оволодівати сучасними знаннями.
- КЗ 8. Здатність оцінювати та забезпечувати якість виконуваних робіт.

Спеціальні (фахові, предметні) компетентності:

- КС 3. Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмноапаратного забезпечення інформаційних

систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС 6. Здатність використовувати сучасні інформаційні системи та технології (виробничі, підтримки прийняття рішень, інтелектуального аналізу даних та інші), методики й техніки кібербезпеки під час виконання функціональних завдань та обов'язків.

КС 7. Здатність застосовувати інформаційні технології у ході створення, впровадження та експлуатації системи менеджменту якості та оцінювати витрати на її розроблення та забезпечення.

Програмні результати навчання

ПРН 2. Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.

ПРН 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

ПРН 6. Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

ПРН 7. Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

Відповідно студенти після засвоєння дисципліни повинні

знати:

- зміст базових понять, предмету та методів курсу;
- основні поняття програмування, принципи розробки програми, основні структури даних;
- типи даних, оператори та керуючі конструкції мови; основні принципи ООП, правила визначення класів в мові Python;

вміти:

- створювати програми, використовуючи засоби операційної системи Windows , середовище розробки PyCharm;

- використовувати змінні та основні типи даних мови програмування Python;
- використовувати процедури введення/виведення;
- використовувати основні вирази мови програмування, зокрема циклів і умовних конструкцій, функцій, генераторів, тощо;
- створювати алгоритми, що вирішують задані завдання і розробляти програми для них мовою Python.

5. Опис навчальної дисципліни

5.1. Загальна інформація

Назва навчальної дисципліни «Програмування мовою Python»											
Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	Семинарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	3	4	120	30	-	-	30	60	-	залік
Заочна	-	-	-	-	-	-	-	-	-	-	-

5.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма						заочна форма						
	усього	у тому числі					усього	у тому числі					
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	
Теми лекційних занять	Змістовий модуль 1. Основи Python.												
Тема 1. Введення в програмування на мові Python	7	2				5	-	-	-	-	-	-	
Тема 2. Елементи функціонального програмування	10	2		2		6	-	-	-	-	-	-	
Тема 3. Модулі: загальна картина	12	2		4		6	-	-	-	-	-	-	
Тема 4. Об'єктноорієнтоване програмування	14	4		4		6	-	-	-	-	-	-	

-Тема 5. Обробка текстів. Регулярні вирази. Unicode	12	4		4		4	-	-	-	-	-	-
Разом за ЗМ1	55	14		14		27	-	-	-	-	-	-
Теми лекційних занять	Змістовий модуль 2. Системне програмування.											
Тема 6. Розробка Web-додатків	17	4		4		9	-	-	-	-	-	-
Тема 7. Робота із Базою Даних	17	4		4		9	-	-	-	-	-	-
Тема 8. Створення додатків з графічним інтерфейсом користувача	17	4		4		9	-	-	-	-	-	-
Тема 9. Інтеграція Python з іншими мовами програмування.	14	4		4		6	-	-	-	-	-	-
Разом за ЗМ 2	65	16		16		33	-	-	-	-	-	-
Усього годин	120	30		30		60	-	-	-	-	-	-

Примітка. Методичні рекомендації та завдання до лабораторних робіт доступні в системах електронного навчання (Moodle/Google Classroom), а також у кафедральному репозиторії за посиланням https://drive.google.com/drive/folders/1t8f3RVEIEe3ZN_kwbK2UTX6ic_hET3lh
Програмне забезпечення для виконання лабораторних робіт: PyCharm, Anaconda, Google Colaboratory.

5.3. Зміст завдань для самостійної роботи

№ з/п	Назва теми	Кількість годин
Змістовий модуль 1. Основи Python.		
1	<i>Введення в програмування на мові Python.</i>	5
2	<i>Елементи функціонального програмування.</i>	6
3	<i>Модулі: загальна картина.</i>	6
4	<i>Об'єктно-орієнтоване програмування.</i>	6
5	<i>Обробка текстів. Регулярні вирази. Unicode.</i>	4
Всього годин за змістовим модулем 1		27
Змістовий модуль 2. Системне програмування.		
1	<i>Розробка Web-додатків.</i>	9
2	<i>Робота із Базою Даних.</i>	9
3	<i>Створення додатків з графічним інтерфейсом користувача.</i>	9
4	<i>Інтеграція Python з іншими мовами програмування.</i>	6
Всього годин за змістовим модулем 2		33
Разом		60

Самостійна робота студента полягає в опрацюванні теоретичного матеріалу, більш глибокому та детальному розгляді окремих питань курсу, командному виконанні лабораторних робіт, підготовці лекційних занять, за бажанням – проходженні курсів відповідної тематики на відомих освітніх платформах (Coursera, Prometheus тощо). Оцінювання самостійної роботи відбувається під час захисту лабораторних робіт (30% оцінки кожної лабораторної роботи складає оцінювання самостійної роботи).

6. Форми і методи навчання

Форми навчання – це лекції-візуалізації (із застосуванням комп'ютерної техніки), проблемні лекції, лабораторні заняття, інтегровані заняття, заняття з використанням систем електронного навчання Moodle/Google Classroom; індивідуальні та групові консультації, самостійна робота; використання елементів дистанційного навчання (за потреби): відеолекції, відеозаняття і відеоконференції засобами Google Meet, Zoom тощо.

Підходи до навчання – використовуються студентоцентрований, проблемно-орієнтований, діяльнісний, комунікативний, професійноорієнтований, міждисциплінарний підходи.

Для викладання навчальної дисципліни використовуються наступні **методи навчання**:

- *пояснювально-ілюстративні* (спрямовані на повідомлення готової інформації різними засобами (словесними, наочними, практичними) та усвідомлення і запам'ятовування цієї інформації студентами);
- *компетентнісний* (навчання, спрямоване на розвиток навичок, умінь і якостей, які знадобляться в професійній діяльності);
- *репродуктивний* (використовується під час практичних і лабораторних занять, а також під час самостійної роботи студентів; передбачає роботу студентів за визначеним алгоритмом);
- *частково-пошукові або евристичні* (організація активного пошуку розв'язання поставлених або самостійно сформульованих пізнавальних завдань, над якими студенти працюють самостійно під керівництвом педагога або на основі евристичних програм та вказівок);
- *проектно-дослідницькі* (групи студентів отримують комплекс завдань чи проблемне питання, визначений час для виконання; метод спрямований на розвиток пошукових, аналітичних якостей студентів, а також навичок командної роботи).

7. Система контролю та оцінювання

Види та форми контролю. Поточний контроль за роботою студентів під час вивчення навчальної дисципліни здійснюється за допомогою наступних методів:

- захист студентами лабораторних робіт та проєктів;
- поточні опитування та тестування.

Проводиться під час лабораторних та практичних занять. Основне завдання – перевірка рівня підготовки студентів за визначеною темою (навчальним елементом). Основна мета – забезпечення зворотного зв'язку між викладачами та студентами, управління навчальною мотивацією студентів. Інформація, одержана при поточному контролі, використовується як викладачем – для коригування методів і засобів навчання, – так і студентами – для планування самостійної роботи.

Підсумковий контроль здійснюється у формі заліку в терміни, встановлені графіком навчального процесу та в обсязі навчального матеріалу, визначеного робочою програмою дисципліни. **Засоби оцінювання:**

- опитування теоретичного матеріалу;
- лабораторні роботи;
- презентації результатів виконаних завдань;
- тестові завдання.

7.1. Критерії оцінювання результатів навчання з навчальної дисципліни

Оцінювання знань студентів з навчальної дисципліни "Програмування мовою Python" здійснюється на основі результатів поточного та підсумкового контролю.

Оцінювання знань студентів здійснюється за 100-бальною шкалою. Результати роботи студентів впродовж навчального семестру оцінюються в ході поточного контролю в діапазоні від 1 до 60 балів (включно), а результати підсумкового контролю (заліку) оцінюються від 1 до 40 балів (включно).

Поточний контроль роботи студентів з навчальної дисципліни "Програмування мовою Python" здійснюється за наступними критеріями:

- виконання та захист лабораторних робіт №1-8 – до 6 балів кожна;
- контрольна робота – до 12 балів.

Для успішного проходження підсумкового оцінювання здобувач освіти має набрати не менше 4 балів за кожен лабораторну та контрольну роботу. Ті студенти, які за результатами поточного контролю отримали не менше 20 балів, допускаються до заліку.

Залік проводиться у вигляді тестування (**40 балів**) і складається з 40 тестових питань. За кожен правильну відповідь студент отримує **1** бал. У сумі з модульними контролями (**60 балів**) це загалом складатиме максимально **100** балів.

Підсумковий контроль здійснюється у формі заліку в терміни, встановлені графіком навчального процесу та в обсязі навчального матеріалу, визначеного робочою програмою дисципліни.

Розподіл балів, які отримують студенти

Поточне оцінювання (<i>аудиторна та самостійна робота</i>)									Кількість балів (залік)	Сумар на к-ть балів
Змістовий модуль 1					Змістовий модуль 2				40	100
T1	T2	T3	T4	T5	T6	T7	T8	T9		
6	6	6	6	10	6	6	6	8		

Підсумкова оцінка. Підсумкова оцінка виставляється за загальною сумою балів, набраних студентом під час модульних контролів та на заліку, згідно із наступною таблицею:

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
80 – 89	B	добре	
70 – 79	C		
60 – 69	D	задовільно	
50 – 59	E		
35 – 49	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0 – 34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

7.2. Умови зарахування результатів неформальної освіти

Результати неформальної освіти можуть бути зараховані студенту згідно з Положенням ЧНУ "Про взаємодію формальної та неформальної освіти, визнання результатів навчання (здобутих шляхом неформальної та/або інформальної освіти, в системі формальної освіти)".

Якщо у студента наявні сертифікати про неформальну освіту з переліку курсів, наведених у переліку інформаційних джерел (джерела 8-9 з п.9), то студентові зараховуються бали за виконання лабораторних робіт, які покриваються пройденим курсом, без додаткової верифікації. Такий студент має лише виконати контрольну роботу та пройти підсумковий контроль. В разі наявності сертифікату про пройдений курс із тематики дисципліни "Програмування мовою Python", якого немає в списку інформаційних джерел,

зарахування результатів відбувається з верифікацією у формі співбесіди студента з лектором.

Також, як виконані види роботи з відповідних тем, студенту можуть бути зараховані бали за наукові публікації у матеріалах науково-практичних конференцій та фахових чи апробаційних виданнях або подання роботи на конкурс студентських наукових робіт.

7.3. Політика курсу

Політика щодо відвідування: відвідування занять є обов'язковим (виключення складають студенти, які навчаються за індивідуальним графіком та ті, кому зараховано результати неформальної освіти). Для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей. Засвоєння теми лекції, пропущеної з поважної причини, перевіряється під час складання підсумкового контролю. Пропуск лекції з неповажної причини відпрацьовується студентом (співбесіда, реферат тощо). Пропущені практичні та лабораторні заняття, незалежно від причини пропуску, студент відпрацьовує згідно з графіком консультацій. За об'єктивних причин (наприклад, хвороба, участь у програмі міжнародного обміну, індивідуальний графік навчання) навчання може відбуватись у змішаній формі (очно-дистанційній) за погодженням із керівником курсу.

Політика академічної доброчесності: обов'язковими є посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерела інформації; списування під час контрольних заходів заборонені (в т. ч. із використанням мобільних пристроїв).

Політика щодо дедлайнів та перескладання: роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (від 10% до -50% від максимальної кількості балів – залежно від терміну затримки здачі роботи). Порушення терміну здачі роботи з поважної причини не призводить до втрати балів. Складання (перескладання) іспиту відбувається за встановленим деканатом розкладом. Якщо студента не допущено до складання іспиту через те, що він набрав менше 20 балів протягом семестру, то до перескладання він має доздати викладачу лабораторні роботи.

Політика щодо оскарження оцінювання: забезпечення об'єктивності та прозорості оцінювання регламентується п.3.8-3.9 Положення ЧНУ "Про контроль і систему оцінювання результатів навчання здобувачів вищої освіти"; оскарження результатів підсумкового оцінювання здійснюється у відповідності до Положення ЧНУ "Про апеляцію на результати підсумкового семестрового контролю знань студентів".

8. Рекомендована література

8.1. Основна

1. Програмування мовою Python: лабораторний практикум. / Укл.: Літвінчук Ю.А., Малик І.В., Горбатенко М.Ю. Чернівці : Чернів. нац. ун-т ім. Ю.Федьковича, 2021. 101 с.
2. Васильєв О. Програмування мовою Python. – Тернопіль: Навчальна книга Богдан, 2019. 204 с.
3. Ерік М. Пришвидшений курс Python. Київ: Видавництво Старого Лева, 2021. 600 с.
4. Eric Matthes. Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming. 2019. 544 p.
5. Mark Lutz. Learning Python. 5th Edition, O'Reilly Media 2013. 1643 p.
6. John M Zelle. Python Programming: An Introduction to Computer Science. Franklin beedle & Assoc. 2016. 552 p.
7. Wes Mckinney. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition, O'Reilly Media 2017. 524 p.
8. Krishna Rungta. Learn Python in 1 Day: Complete Python Guide with Examples. Guru99, 2016. 213 p.

8.2. Допоміжна

1. Naomi Ceder The Quick Python Book 3rd Edition. NY: Manning Publications Co., 2018. 432 p.
2. Kenneth A. Lambert Fundamentals of Python: first programs. NY: Cengage Learning, 2018. 476 p.
3. Mark L. Learning Python, 5th Edition. Sebastopol: O'Reilly Media, 2019. 648 p.

9. Інформаційні ресурси 4.

<https://younglinux.info/python>.

5. <https://pro.arcgis.com/ru/pro-app/arcpy/main/arcgis-pro-arcpy-reference.htm>.

6. <http://pythonguide.rozh2sch.org.ua/>.

7. Guido van Rossum and Fred L. Drake, jr. Python Documentation.
<https://www.python.org/doc/>.

8. <https://www.coursera.org/learn/python>

9. <https://prometheus.org.ua/prometheus-plus/python-beetroot-course/>