

# ОСНОВИ ПРОГРАМУВАННЯ PYTHON

## СИЛАБУС

<b>Шифр і назва спеціальності</b>	121 Інженерія програмного забезпечення, 122 Комп'ютерні науки, 126 Інформаційні системи та технології	<b>Інститут / факультет</b>	Факультет комп'ютерних наук і програмної інженерії
<b>Назва програми</b>	«Інженерія програмного забезпечення» «Комп'ютерні науки та інтелектуальні системи» «Програмне забезпечення інформаційних систем»	<b>Кафедра</b>	Кафедра програмної інженерії та інформаційних технологій управління
<b>Тип програми</b>	Освітньо-професійна	<b>Мова навчання</b>	Українська, Англійська

## Викладач

**ПІБ, електронна пошта**

Козуля Марія Михайлівна, mariia.kozulia@khp.edu.ua



Кандидат технічних наук (21.06.01 - екологічна безпека), доцент кафедри програмної інженерії та інформаційних технологій управління. Досвід роботи – з 2016 року. Автор (співавтор) понад 75 наукових та навчально-методичних публікацій (h-index= 6, i10-index= 2 in Google Scholar - <https://scholar.google.ru/citations?user=tRyBDzQAAAAJ&hl=ru>; ORCID <https://orcid.org/0000-0002-4090-8481>). Основні курси: «Основи програмування Python» (лекції та лабораторні заняття), «Поглиблений курс програмування Python» (лекції та лабораторні заняття), «Теорія прийняття рішень» (лекції та лабораторні заняття), «Моделі та методи теорії прийняття рішень» (лекції та лабораторні заняття), «Грін комп'ютинг» (лекції та лабораторні заняття).

## Загальна інформація про курс

<b>Анотація</b>	Курс «Основи програмування Python» є навчальною дисципліною з циклу вибіркової підготовки за спеціальністю 121 «Інженерія програмного забезпечення», 122 Комп'ютерні науки, 126 Інформаційні системи та технології. Вона викладається у третьому семестрі в обсязі 180 годин (6 кредити ECTS), зокрема: лекції – 32 годин, лабораторні заняття – 32 години, самостійна робота – 116 години. Індивідуальних завдань не передбачено. Вивчення дисципліни завершується заліком.
<b>Цілі курсу</b>	Формування у студентів теоретичних та практичних знань з основ програмування на мові Python, оволодіння методами збору і обробки даних.
<b>Формат</b>	Лекції, лабораторні заняття. Поточний контроль – лабораторні роботи, проміжний модульний контроль. Підсумковий контроль – залік.
<b>Семестр</b>	3

**Обсяг (кредити) / Тип курсу**  
(обов'язковий / вибіркового)

6/ Вибірковий

**Лекції (години)**

32

**Лабораторні**

32

**Самостійна робота**

116

		заняття (години)	(години)
<b>Програмні компетентності</b>	<p>121-K01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>121-K02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>121-K05. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>121-K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>121-K15. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>121-K16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.</p> <p>121-K19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.</p> <p>121-K26 Здатність до алгоритмічного та логічного мислення</p> <p>122-ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>122-ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>122-ЗК6. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>122-ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>122-ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p>122-СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.</p> <p>122-СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p> <p>126-K31. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>126-K32. Здатність застосовувати знання у практичних ситуаціях.</p> <p>126-K3 5. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>126-K3 6. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел.</p> <p>126-K3 8. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p>126-КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).</p> <p>126-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.</p> <p>126-КС 8. Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу.</p>		
	<b>Результати навчання</b>	<b>Методи викладання та навчання</b>	<b>Форми оцінювання (поточне оцінювання CAS, підсумкове оцінювання FAS)</b>
<p>121-ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>121-ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні</p>	<p>Міні-лекція «Вступ до Python. Типи даних. Введення та виведення даних»</p> <p>Міні-лекція «Умовна конструкція if..else, if...elif...else. Модуль Math у Python»</p> <p>Проблемні лекції. «Анонімні функції, інструкції lambda. Функції-генератори. Документування коду»</p> <p>Проблемна лекція. «Інкапсуляція. Геттери.</p>	<p><b>Поточне оцінювання CAS:</b></p> <p>Оцінювання роботи студентів у процесі лабораторних занять</p> <p>Проміжний модульний контроль</p> <p><b>Підсумкове оцінювання FAS:</b></p>	

<p>принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>121-ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p> <p>121-ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>121-ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p> <p>121-ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>121-ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p> <p>122-ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.</p> <p>122-ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.</p> <p>122-ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).</p> <p>122-ПР13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.</p> <p>122-ПР20. Розробляти архітектуру програмних систем та їх окремих компонент при побудові інтелектуальних</p>	<p>Сетери. Успадкування. Асоціація. Слабкі посилання.» Проблемна лекція. «Поліморфізм. Ітератори. Метакласи. Мультиметоди.» Проблемна лекція. «Потоки. Робота з потоками. Мультипоточність. Демони.» Проблемна лекція. «Умови гонки. Синхронізація потоків.»</p>	<p>Залік</p>
---	--	--------------

систем управління у різних галузях, а також управляти процесами життєвого циклу програмного забезпечення інтелектуальних систем управління.

126-ПР 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.

126-ПР 4. Проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.

126-ПР 5. Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.

126-ПР 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.

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

## СИСТЕМА ОЦІНЮВАННЯ

Розподіл балів для оцінювання успішності студента	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	Нарахування балів	<b>100% підсумкове оцінювання</b> у вигляді заліку (30%) та поточного оцінювання (70%). <b>30% залік</b> <b>70% поточне оцінювання:</b> Модуль №1 (10%) Модуль №2 (20%) Лабораторні роботи (40%)
	90-100	A	відмінно		
	82-89	B	добре		
	74-81	C			
	64-73	D	задовільно		
60-63	E				

	35-59	FX	незадовільно з можливістю повторного складання		
	0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни		

**Політика курсу** Студент зобов'язаний відвідувати всі заняття згідно навчального розкладу та дотримуватися норм академічної етики. Для вивчення дисципліни необхідно мати власний персональний комп'ютер та/або використовувати комп'ютери обчислювального центру кафедри. Студент повинен працювати з обов'язковою та додатковою літературою, зокрема з інформаційними ресурсами в Інтернеті. Усі лабораторні роботи мають бути виконані та здані студентом протягом семестру, у якому викладається дисципліна, до початку залікового тижня. Без особистої присутності студента підсумковий контроль не проводиться.

### Структура та зміст курсу

Лекція	Тема лекції	Лабораторна робота	Тема лабораторної роботи	Самостійна робота
Лекція 1	Вступ до Python. Типи даних. Введення та виведення даних	Лабораторна робота 1	Початок роботи з Python. П350	Динамічна робота з типа даних.
Лекція 2	Умовна конструкція if..else, if...elif...else. Модуль Math у Python.	Лабораторна робота 2	Робота з умовними операторами, умовна конструкція if..else, if...elif...else. Модуль Math у Python. П350	Виключення, особливості роботи та область їх використання.
Лекція 3	Список, Рядок, кортеж, діапазон та робота з ними. Цикл for, while. List compression.	Лабораторна робота 3	Робота з даними: список, рядок, кортеж, діапазон, множини, словники та робота з ними. П350	Особливості створення метакласів та їх використання.
Лекція 4	Множини, словники та робота з ними.	Лабораторна робота 4	Робота з циклами. П350	Потоки. Область використання та особливості використання.
Лекція 5	Область видимості змінних. Функції. Анонімні функції, інструкції lambda. Функції-генератори. Рекурсія. Документування коду.	Лабораторна робота 5	Робота з функціями, виключеннями та класами. П351	Інструменти інтеграційного та системного тестування.
Лекція 6	Виключення. Відладка за допомоги інструкції assert.	Лабораторна робота 6	Робота з вбудованими модулями. П351	
Лекція 7	Модулі: створення та подальше їх використання. Робота з часом. Робота з датою.	Лабораторна робота 7	ООП (Object-Oriented programming). П352	
Лекція 8	Пакети: імпорт, створення, подальше використання. Огляд стандартних бібліотек. Шаблони регулярних виразів.	Лабораторна робота 8	Робота з файлами. П352	
Лекція 9	Вступ до об'єктно-орієнтованого програмування. Створення класу та робота з ним. Magic Methods.	Лабораторна робота 9	Робота з потоками. Тестування. П352	
Лекція 10	Декоратори. Інкапсуляція. Геттери. Сетери. Успадкування. Асоціація. Слабкі посилання.			

<b>Лекція 11</b>	Поліморфізм. Ітератори. Метакласи. Мультиметоди.				
<b>Лекція 12</b>	Файли. Робота з файлами. Копіювання файлів.				
<b>Лекція 13</b>	Потоки. Робота з потоками. Мультиточність. Демони.				
<b>Лекція 14</b>	Умови гонки. Синхронізація потоків.				
<b>Лекція 15</b>	Асинхроність.				
<b>Лекція 16</b>	Написання тестів				

### Література

<b>Обов'язкова</b>	1. Васильев, А. Н. (2019). Python на примерах : практический курс по программированию. 3-е изд. Санкт-Петербург: Наука и техника, 430 с. 2. Доусон, М. (2014). Програмуємо на Python. Санкт-Петербург: Питер, 416 с. 3. Лутц, М. (2011). Изучаем Python, 4-е изд.: пер. с англ. Санкт-Петербург: Символ-Плюс, 1280 с. 4. Лутц, М. (2011). Программирование на Python, 4-е изд.: пер. с англ. Санкт-Петербург: Символ-Плюс. Т. I, 992 с. 5. Лутц, М. (2011). Программирование на Python. 4-е изд.: пер. с англ. Санкт-Петербург: Символ-Плюс. Т. II, 992 с.	<b>Додаткова</b>	1. Прохоренко, Н. А. (2012). Python 3 и PyQt. Разработка приложений. Санкт-Петербург: БХВ-Петербург, 704 с. 2. Вабищевич, П. Н. Численные методы. Вычислительный практикум, 320 с. 3. Пилгрим Марк. Погружение в Python 3 (Dive into Python 3 на русском). 4. Прохоренко, Н. А. (2011). Самое необходимое. Санкт-Петербург: БХВ-Петербург, 416 с. 5. Хахаев, И. А. (2010). Практикум по алгоритмизации и программированию на Python. Москва: Альт Линукс, 126 с. (Библиотека ALT Linux). 6. Чаплыгин, А. Н. Учимся программировать вместе с питоном. 7. Шапошникова, С. Основы программирования на Python. Вводный курс. 8. Deitel H.M. et al. Python. How to Program. 9. Allen Downey. ThinkPython+Kart[Python_3.2].
	<b>ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ</b> 1. Beginner's Guide to Python. [Electronic resource]. Access mode: <a href="https://wiki.python.org/moin/BeginnersGuide">https://wiki.python.org/moin/BeginnersGuide</a> 2. The Python Standard Library. [Electronic resource]. Access mode: <a href="https://docs.python.org/3/library/index.html">https://docs.python.org/3/library/index.html</a>		

### Норми академічної етики

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність

Силабус за змістом повністю відповідає робочій програмі курсу.