



# ПРОГРАМУВАННЯ МОВОЮ PYTHON

## Робоча програма навчальної дисципліни (Силабус)

### 1. Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>11 Математика та статистика</i>
Спеціальність	<i>113 Прикладна математика</i>
Освітня програма	<i>Наука про дані та математичне моделювання</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>1 курс, осінній семестр</i>
Обсяг дисципліни	<i>3,5 кредити</i>
Семестровий контроль/ контрольні заходи	<i>Залік/Контрольні (3) та практичні (2) роботи</i>
Розклад занять	<i>Лекції — 1 раз на 2 тижні (9 лекцій), практики — 1 раз на тиждень (18 занять)</i>
Мова викладання	<i>Українська/Англійська</i>
Інформація про керівників курсу / викладачів	Лектор: <i>Борисенко Павло Борисович, <a href="mailto:pavlo.borysenko@gmail.com">pavlo.borysenko@gmail.com</a></i> Практичні: <i>Борисенко Павло Борисович, <a href="mailto:pavlo.borysenko@gmail.com">pavlo.borysenko@gmail.com</a>;</i> <i>Щьоголев Максим Олегович, <a href="mailto:shchogolev.maksym@gmail.com">shchogolev.maksym@gmail.com</a>;</i> <i>Гожий Віктор Вікторович, <a href="mailto:viktor.hozhyi@gmail.com">viktor.hozhyi@gmail.com</a></i>
Розміщення курсу	Канали <i>#python-course</i> та <i>#python-laboratory</i> у кафедральному Slack Лекції: <a href="https://www.youtube.com/playlist?list=PLjBFeyMhtyMsEAamVDOWafp1bhzITRRYs">https://www.youtube.com/playlist?list=PLjBFeyMhtyMsEAamVDOWafp1bhzITRRYs</a>

### 2. Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

*Уміння програмувати є невід'ємним елементом підготовки фахівців у сфері роботи з даними. Мова Python є де-факто головним інструментом для обробки даних завдяки своїй синтаксичній простоті та повсюдності поширення, наявності великої кількості існуючих модулів та бібліотек, а також підтримці з боку великих корпорацій та бізнесу, великому міжнародному ком'юніті користувачів. Крім того, Python є хорошим інструментом опанування основ програмування, завдяки своїй лаконічності та строгій структурі коду.*

*Курс «Програмування мовою Python» є базовим курсом підготовки спеціалістів у сфері науки про дані та математичного моделювання.*

*Метою курсу є:*

- вивчення основних понять і концепцій програмування,*
- базових конструкцій та синтаксису мови Python,*
- принципів та підходів до вирішення алгоритмічних задач.*

*Предметом вивчення є програмний код мовою Python та його синтаксичні примітиви (умови, цикли, функції тощо); алгоритми, записані у вигляді коду.*

*Після засвоєння дисципліни студенти матимуть наступні:*

- компетентності:
  - здатність до абстрактного мислення, аналізу та синтезу;
  - знання та розуміння предметної області та розуміння професійної діяльності;
  - здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію;
  - здатність використовувати сучасні технології програмування та тестування програмного забезпечення;
  - здатність зрозуміти постановку завдання, сформульовану мовою певної предметної галузі, здійснювати пошук та збір необхідних вихідних даних;
- знання:
  - синтаксису мови Python;
  - основних конструкцій, характерних більшості мов програмування: циклів, умов, операторів, функцій;
  - про типи даних та їх представлення у пам'яті комп'ютера;
  - принципів роботи систем контролю версій;
  - основних команд системи контролю версій git;
  - принципів рекурсії;
  - принципів роботи генераторів та декораторів;
- уміння:
  - автоматизувати вирішення задач;
  - розробляти алгоритми для вирішення задач та представляти їх у вигляді програмного коду;
- навички:
  - роботи з терміналом;
  - використання систем контролю версій;
  - написання інтерактивних програм;
  - роботи із Visual Studio Code та Jupyter;
- досвід:
  - роботи із системою контролю версій git;
  - роботи із віддаленими репозиторіями на GitHub/GitLab;
  - програмування мовою Python;
  - отримання даних із джерел різного формату.

## **2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

*Дисципліна вивчається у осінньому семестрі 1 курсу та базується лише на базових шкільних знаннях з математики та англійської мови.*

*Успішне проходження курсу дозволить студентам продовжити вивчення наступних дисциплін з навчального плану підготовки бакалаврів:*

- Автоматизоване тестування програмного забезпечення
- Алгоритми і структури даних
- Алгоритмічні основи обчислювальної геометрії та комп'ютерної графіки
- Аналіз даних
- Бази даних та інформаційні системи
- Випадкові процеси
- Криптографічні методи захисту інформації
- Математичне моделювання
- Методи оптимізації
- Об'єктно-орієнтоване програмування
- Функційне програмування

*та дисциплін з навчального плану підготовки магістрів:*

- *Інтелектуальний аналіз даних*
- *Машинне навчання*

### **3. Зміст навчальної дисципліни**

*Розділ 1. Базові конструкції Python. Типи даних.*

*Тема 1.1. Програмування як концепція, алгоритми.*

*Тема 1.2. Числові типи даних. Представлення даних у пам'яті. IEEE 754.*

*Тема 1.3. Списки та індексація. Зрізи. Множини, кортежі та словники.*

*Тема 1.4. Рядки і представлення текстової інформації. Кодування. Unicode.*

*Розділ 2. Функції та структура програм. Рекурсія.*

*Тема 2.1. Функції. Лямбда-функції та функції вищих порядків.*

*Тема 2.2. Генератори. Декоратори.*

*Тема 2.3. Рекурсія.*

*Розділ 3. Спілкування програм із зовнішнім світом.*

*Тема 3.1. Введення-виведення та робота з файлами.*

*Тема 3.2. Модулі та організація коду.*

### **4. Навчальні матеріали та ресурси**

#### **Підручники**

1. *Scott Chacon, Ben Straub, Pro Git, 2014.*

*Українська та англійська версії доступні за посиланням: <https://git-scm.com/book/uk/v2>  
Розділи 2 і 6 — обов'язкові. Розділ 3 — бажаний.*

2. *John Hunt, A Beginners Guide to Python 3 Programming, 2019.*

3. *Kent D Lee, Python Programming Fundamentals, 2 ed., 2014.*

#### **Додаткові матеріали**

1. *Документація Python*

*Посилання: <https://docs.python.org/>*

*Особливу увагу варто приділити розділам:*

- *Python Setup and Usage,*
- *Python HOWTOs,*
- *Language Reference.*

2. *Learn Git Branching.*

*Посилання: <https://learngitbranching.js.org/?locale=uk>*

*Це хороший інструмент для тренування принципів створення гілок у git.*

3. *Visualizing Git.*

*Посилання: <http://git-school.github.io/visualizing-git/>*

*Інструмент візуалізації роботи з git.*

4. *Git-It*

*Посилання: <https://github.com/jlord/git-it-electron#what-to-install>*

*Набір інтерактивних задач для опанування git.*

#### **Рекомендовані онлайн-курси**

##### **Базові**

3. *Prometheus. Programming 101*

*[https://courses.prometheus.org.ua/courses/KPI/Programming101/2015\\_T1/about](https://courses.prometheus.org.ua/courses/KPI/Programming101/2015_T1/about)*

4. *Coursera. Python 3 Programming*

*<https://www.coursera.org/specializations/python-3-programming>*

5. Coursera. Google IT Automation  
<https://www.coursera.org/professional-certificates/google-it-automation>
6. Coursera. Python. Основы программирования  
<https://www.coursera.org/learn/python-osnovy-programmirovaniya>
7. EdX. Introduction to Python Programming  
<https://www.edx.org/professional-certificate/introduction-to-python-programming>
8. EdX. Computational Thinking Using Python  
<https://www.edx.org/xseries/mitx-computational-thinking-using-python>

#### Просунуті

9. Coursera. Data Science Python  
<https://www.coursera.org/specializations/data-science-python>
10. EdX. Python Data Science  
<https://www.edx.org/professional-certificate/python-data-science>
11. Coursera. Программирование на Python  
<https://www.coursera.org/specializations/programming-in-python>

## 12. Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компонента)

#### Лекції

##### Розділ 1. Базові конструкції Python. Типи даних

Лекція 1. Програмування як концепція, алгоритми.

Приклади алгоритмічних задач. Змінні та типи даних. Базовий синтаксис Python. Приклади простих програм на Python.

Лекція 2. Числові типи даних. Представлення даних у пам'яті. IEEE 754.

Лекція 3. Списки та індексація. Зрізи. Множини, кортежі та словники. Деякі корисні базові функції.

Лекція 4. Рядки і представлення текстової інформації. Кодування. Unicode.

##### Розділ 2. Функції та структура програм. Рекурсія

Лекція 5. Функції. Лямбда-функції та функції вищих порядків. Вкладені функції та видимість змінних. Передача аргументів. Сортування. Тип None.

Лекція 6. Генератори. Декоратори.

Лекція 7. Рекурсія. Хвостова рекурсія. Еквівалентність рекурсії і циклів.

##### Розділ 3. Спілкування програм із зовнішнім світом

Лекція 8. Введення-виведення. Робота з файлами. Типи файлів.

Лекція 9. Модулі та організація коду. Прикінцеві зауваження.

#### Практики

##### Розділ 0. Інструменти розробки

Практика 1. Встановлення Пайтону. Середовище розробки. Менеджер пакетів pip. Віртуальні середовища venv.

Практика 2. Git. Встановлення та налаштування. Репозиторії. Примітивні операції на репозиторіях: create, add, commit.

Практика 3. GitHub. Поняття віддаленого репозиторію. Створення акаунту. Операції на віддалених репозиторіях: clone, push, pull. Пулл-реквести та git flow.

## Розділ 1. Базові конструкції Python. Типи даних

Практика 4. Перші програми на Пайтоні. Виведення даних на екран і отримання даних з терміналу.

Практика 5. Створення списків: list, генератори, [...]. Маніпуляції списками: пошук, вставлення, видалення, сортування і т.д. Цикли.

Практика 6. Робота з кортежами, множинами і словниками. Приведення одних типів до інших. Копіювання об'єктів.

Практика 7. Робота з рядками. Форматовані рядки. Перетворення між списками і рядками. Байтові рядки.

Практика 8. Валідація вводу та перевірка даних. Assert і try/except. Регулярні вирази

## Розділ 2. Функції та структура програм. Рекурсія

Практика 9. Функції: написання і виклик. Вкладені функції. Композиції функцій.

Практика 10. Лямбда-функції. Функції вищих порядків. map і reduce.

Практика 11. Генератори і використання yield. Нескінченні послідовності. Генератори і списки.

Практика 12. Декоратори. Декоратори з параметрами. Використання декораторів для перевірки пре- і пост-умов, логування.

Практика 13. Рекурсивні функції. Рекурсія з акумулятором і без. Непряма рекурсія.

Практика 14. Рекурсивні структури даних. Заміна циклів на рекурсію і навпаки

## Розділ 3. Спілкування програм із зовнішнім світом

Практика 15. Робота з файлами. Читання і запис текстових файлів. Бінарні файли і серіалізація даних. with-синтаксис.

Практика 16. Модулі для роботи зі спеціальними форматами файлів. Читання csv і JSON.

Практика 17. Створення модуля. Спеціальний модуль \_\_main\_\_

Практика 18. Спеціальні теми.

### 6. Самостійна робота студента

До самостійного опрацювання виносяться:

- підготовка до аудиторних та практичних занять — до 2 годин на тиждень;
- підготовка до контрольних робіт та заліку — до 10 годин за семестр;
- вивчення наступних тем:
  - Синтаксичні одиниці Python (ключові слова, оператори).
  - Теоретичні основи роботи систем контролю версій (git).
  - Основні команди системи контролю версій git.
  - Уміст та структура базових модулів Python
  - Додаткові функції модулю csv.
- виконання 2 домашніх практичних робіт — до 30 годин кожна.

## 13. Політика та контроль

### 7. Політика навчальної дисципліни

#### Відвідування

Відвідування лекцій необов'язкове, але ми заохочуємо студентів не пропускати лекційні заняття через можливість ставити уточнюючі питання та брати участь у живому обговоренні. Записи

лекцій (та оглядової частини практичних занять) поточного чи минулих років будуть доступні онлайн.

Відвідування практичних занять обов'язкове. Активність на практичних заняттях становить 45% семестрового рейтингу, тому не варто нехтувати нею. Протягом семестру допускається пропуск до 3 практичних занять без втрат для семестрового рейтингу. Практичні заняття пропущені з поважних причин можуть бути замінені на виконання додаткових завдань за погодженням із викладачем.

Пропущені контрольні роботи перездавати не можна.

### **Оцінювання**

Оцінювання контрольних та практичних робіт відбувається з точністю до десятих, округлення за звичними правилами.

### **Дедлайни**

Контрольні роботи мають бути здані у рамках часу, відведеного на їх проведення.

Практичні роботи мають бути здані в рамках відповідного заняття. Іноді, викладач, відповідальний за заняття, буде продовжувати цей термін — зазвичай до кінця календарного дня, у який проводиться заняття.

Домашні практичні роботи мають окремі визначені терміни (дедлайни). Роботи, здані після цих термінів, будуть оцінюватися з модифікатором:

- здані після **софт дедлайну** (зазвичай місяць після отримання завдання) — 0,5, тобто отримують половину балів;
- здані після **хард дедлайну** (тиждень до заліку) — не отримують балів.

### **Додаткові бали**

Активність на лекціях та практичних заняттях — відповіді на запитання викладача, знаходження помилок у лекційних чи практичних матеріалах; питання, що свідчать про вдумливу роботу з навчальним матеріалом, надання оригінальних рішень практичних завдань чи домашніх практичних робіт тощо — заохочується додатковими балами на розсуд викладача.

Крім того, заохочується додатковими балами підтверджене сертифікатами проходження курсів (онлайн чи офлайн), що стосуються тем дисципліни.

Протягом курсу можна отримати не більше 10 додаткових балів.

### **Академічна доброчесність**

Ми підтримуємо принципи академічної доброчесності і рівності всіх студентів. У випадку виявлення випадків списування (у контрольних, практичних, домашніх роботах) чи плагіату — бали за відповідні роботи будуть анульовані. Повторні порушення принципів академічної доброчесності можуть призвести до недопуску до складання заліку.

Викладачі можуть перевіряти роботи, виконані у рамках курсу, за допомогою систем виявлення плагіату Unicheck та MOSS.

Якщо не зазначено іншого, усі контрольні заходи проводяться у форматі «відкритої книги». Це означає, що ви маєте право користуватися будь-якими ресурсами, окрім допомоги сторонніх осіб. Ми довіряємо нашим студентам і покладаємо надію на те, що вони не порушать цю довіру.

## **8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)**

Поточний контроль:

- **модульні контрольні роботи (15%):**  
5 балів x 3 роботи = **15 балів**

Модульні контрольні роботи представляють собою практичне завдання за темою модуля. На виконання кожної модульної контрольної роботи виділяється до 30 хвилин. Ці роботи проводяться на початку чи в кінці лекцій. Вас буде заздалегідь попереджено про проведення контрольної роботи.

- **робота на практичних заняттях (45%):**

3 бали x 15 занять = **45 балів**

У розрахунок беруться 15 занять із максимальними балами із 18 можливих.

Зазвичай, на кожному практичному занятті розглядається 3 задачі по 1 балу кожна.

- **домашні практичні роботи (40%):**

20 балів x 2 роботи = **40 балів**

Домашні практичні роботи складаються із 8-10 завдань на написання чи доповнення/редагування коду, кожне з яких оцінюється в 1, 2 чи 3 бали. Після перевірки роботи, у студента є можливість один раз виправити помилки для підвищення оцінки.

Домашні практичні роботи мають дедлайни як описано вище.

**Календарний контроль:** проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Календарний контроль проводиться за результатами практичних занять, проведених на момент початку контролю.

**Семестровий контроль:** залік.

Залікова оцінка виставляється на основі семестрового рейтингу, або — за бажанням студента чи при семестровому рейтингу менше 60 балів — за результатами написання залікової контрольної роботи.

Залікова контрольна робота складається із 10 практичних питань 4 рівнів складності (по 3 питання рівнів 1-3 та одного питання рівня 4), що оцінюються у 5, 10, 15 і 20 балів за кожне відповідно.

Умови допуску до семестрового контролю: **мінімальний рейтинг не нижче 25 балів та виконання принаймні 1 домашньої практичної роботи.**

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

## 9. Додаткова інформація з дисципліни (освітнього компонента)

- **додатки до силабусу** — приклади залікової контрольної роботи, положення про РСО;
- **зазвичай залік проходить на останньому занятті з дисципліни, відповідно, хард дедлайн буде за тиждень до цього;**
- **у випадку проведення курсу дистанційно, результати контролю (контрольні та залікові роботи) мають бути виконані в цифровому вигляді: як текстові файли, чи файли з кодом, або за неможливості — у вигляді розбірливих фото. Усі такі матеріали мають бути завантажені на указаний викладачем ресурс у терміни відведені під відповідний тип контролю. Ми надаватимемо буферні 5 хвилин на випадок форс-мажорних подій. У окремих випадках, залікова контрольна може бути проведена у формі співбесіди.**

**Робочу програму навчальної дисципліни (силабус):**

Складено асистентом Борисенком Павлом Борисовичем

Ухвалено кафедрою ПМА (протокол № 13 від 16.06.2022)

Погоджено Методичною комісією факультету прикладної математики (протокол № 9 від 24.06.2022)