

1. ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА

Прізвище, ім'я, по батькові	Плакасова Жаннетта Миколаївна
Науковий ступінь	
Наукове звання	
Посада	ст. викладач кафедри ПЗАС
Місце роботи	ЧДТУ
Адреса кафедри	18006, м. Черкаси, бул. Шевченка 460, ауд. 605 першого корпусу.
Контактний телефон	(0472) 51-15-87
Профайл викладача	https://chdtu.edu.ua/fitis/kpzas/staff/item/2614-plakasova-zhannetta-mykolaiivna
e-mail:	Zh.plakasova@chdtu.edu.ua
Профайл дисципліни	http://fitis.moodle.chdtu.edu.ua/course/view.php?id=634
Розклад консультацій	Вівторок ауд. 605-1, 12-00

2. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Загальні характеристики		Навчальне навантаження з дисципліни	
		денна форма навчання	заочна форма навчання
Вибірков,а циклу загальної підготовки		Курс підготовки:	
		2	2
Загальна кількість кредитів ЄКТС	4	Семестр підготовки:	
Загальна кількість годин	120	4	4
Кількість аудиторних годин	48/54	Лекції	
		18	4
Кількість годин самостійної роботи	72/66	Практичні, семінарські	
		-	-
Мова навчання - українська		Лабораторні	
		36	4
		Самостійна робота	
		66	112
		Форма підсумкового контролю	
		залік	залік

3 МЕТА І ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета викладання дисципліни	<p>Вивчення мови і її особливостей настільки, наскільки це дозволяє вирішити прикладні завдання, що виникають в процесі вивчення тієї чи іншої теми.</p> <p>Набуття ключових компетентностей, теоретичних знань і практичних навичок з програмування в Інтернет у різних сферах професійної діяльності.</p>
Завдання вивчення дисципліни	<ul style="list-style-type: none"> - оволодіння основними поняттями програмування мовою Python; - ознайомлення з новітніми інформаційними технологіями програмування алгоритмічною мовою Python; - розуміння концепцій, загальних для всіх веб-API; - набуття практичних навичок використання методів і засобів програмування алгоритмічною мовою Python.

4 РЕЗУЛЬТАТИ НАВЧАННЯ

№ з/п	Результати навчання
1.	Вивчення Python 3 як мультипарадигменного мовного засобу
2.	Підключати додаткові модулі, стандартні модулі і пакети Python 3;
3.	Розробляти читабельні програми на мові програмування Python 3;
4.	Використовувати як вбудовану, так і доступну в Мережі документацію по бібліотеках Python 3;
5.	Розуміння концепцій, загальних для всіх веб-API
6.	Створювати додаткові модулі і пакети на Python 3 для основної програми;
7.	Програмувати прості API на мові Python 3

5 ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

<p>Змістовий модуль №1 <i>Назва модуля Особливості Python</i></p>
<p>Тема 1 <i>Назва теми</i> Структура Python-програм</p>
<p><i>Зміст теми</i> Структура Python-програм. Лексична структура . Коментарі . Літерали. Знаки пунктуації. Ідентифікатори.</p>

Ключові слова. Синтаксична структура. Складові елементи python-програми. Структура файлу і кодування програми. Виконання Python-програм. Установка і запуск Python. Установка Ide, PyCharm.
Тема 2 Назва теми Типи даних. Загальні функції.
Зміст теми Поняття типу даних і змінної. Класифікація типів даних. Оператор присвоєння. Управління пам'яттю і збирач сміття. Загальні функції. Перевірка типів. Взаємне перетворення. Пріоритет операцій. Поверхнєве і глибоке копіювання. Константи. Сортування. Введення / виведення.
Тема 3 Назва теми Потік команд (керуючі структури)
Умовний оператор. Цикли. Цикл з умовою. Спільний цикл (цикл по колекціях). Ітератори. Деякі техніки переміщення по колекціях. Переривання і продовження циклів. Комбінація циклів і умов. Колекційні включення
Змістовий модуль №2 <i>Назва модуля Підпрограми</i>
Тема 4 Назва теми Основні поняття і механізм роботи.
Визначення підпрограми. Виклик підпрограми. Механізм роботи. Переваги і недоліки. Функції в Python. Глобальні і локальні функції. Параметри і аргументи. Позиційні і ключові параметри / аргументи. Упаковка і розпакування аргументів. Область видимості. Повернення кількох значень Рекурсія. Рядки документації. Анонімні функції. Умовний оператор. Цикли. Цикл з умовою. Спільний цикл (цикл по колекціях)
Змістовий модуль №3 <i>Назва модуля Мережеве та Інтернет-програмування</i>
Тема Передача даних, стек протоколів TCP / IP, сокети.
TCP / IP. Сокети, IP-адреса і порт. Прикладний рівень стека TCP / IP. Інтернет і Веб. HTTP, HTTPS і URI. HTTP: Типова взаємодія. HTTP: Структура протоколу. HTTP: Управління станом і сесія. HTTP: Приклад взаємодії. API веб-сервісів. Підтримка стека TCP / IP в Python

6 ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№ теми	Назва модулів і тем	Форми організації навчання, кількість годин						Література, інформаційні ресурси
		Денна форма			Заочна форма			
		Лекції	Практичні, лабораторні	Самостійна робота	Лекції	Практичні, лабораторні	Самостійна робота	

Змістовий модуль №1 Особливості Python								
1	Структура Python-програм	2	4	5	2	2	12	1,2,3
2	Типи даних. Загальні функції.	2	4	5			12	1,2,3
3	Потік команд (керуючі структури)	2	4	5			12	1,2,3
Змістовий модуль №2 Підпрограми								
4	Основні поняття і механізм роботи. Функції в Python. Глобальні і локальні функції.	2	4	8			12	1,2,3
	Рекурсія. Рядки документації. Анонімні функції	2	4	8			12	1,2,3
	Умовний оператор. Цикли. Цикл з умовою. Спільний цикл (цикл по колекціях)	2	4	8			12	1,2,3
Змістовий модуль №3 Мережеве та Інтернет-програмування								
4	Передача даних, стек протоколів TCP / IP, сокети.	2	4	9	2	2	10	4,5
	Інтернет і Веб. HTTP, HTTPS і URI. HTTP: Типова взаємодія. HTTP: Структура протоколу. HTTP: Управління станом і сесія.	2	4	9			15	4,5
	HTTP: Приклад взаємодії. API веб-сервісів . Підтримка стека TCP / IP в Python	2	4	9			15	4,5
	Разом	18	36	66	4	4	112	

7 ПРАКТИЧНІ / СЕМІНАРСЬКІ ЗАНЯТТЯ, ЛАБОРАТОРНІ РОБОТИ

№ з/п	Тема заняття / Назва лабораторної роботи	Кількість годин	
		Денна	Заочна

1.	Встановлення та налаштування Python. Додаткові пакети. IDE. PyCharm	4	2
2.	Числа. Рядки. Структури Даних	4	
3.	Умовний оператор. Цикл з умовою	4	
4.	Спільний цикл (цикл по колекціях). Переривання і продовження циклів. Комбінація циклів і умов.	4	
5.	Глобальні і локальні функції. Упаковка і розпакування аргументів. Рекурсія	4	
6.	Модули и пакеты. Установка і використання модулів / пакетів. Створення модулів. QR-коди.	4	
7.	Стандартна бібліотека. Установки додатку. Гра «набери найбільшу суму». Звіт про клієнтів.	4	
8.	Мережеве та Інтернет-програмування	8	2

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

1. Методичні рекомендації до лабораторних робіт

<https://drive.google.com/drive/folders/0B5LzEDMnWOAPNTVHN29lM1JSUWc>

8 САМОСТІЙНА РОБОТА

8.1 Рекомендації до самостійної роботи здобувачів вищої освіти денної форми навчання.

Ряд тем не був включений в даний курс, однак являють собою не менший інтерес і не меншу значимість для розробки додатків.

Додаткові теми.

Розробка графічного інтерфейсу:

— настільні додатки

— веб-додатки

Програмування баз даних.

Процеси і потоки.

Рефакторинг і оптимізація коду.

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

1. Методичні рекомендації до самостійної роботи

1. <https://drive.google.com/drive/folders/0B5LzEDMnWOAPNTVHN29lM1JSUWc>

2. <https://github.com/topics/python>

8.2 Рекомендації до самостійної роботи здобувачів вищої освіти заочної форми навчання.

Опис необхідних робіт чи завдань

1. Комплексна задача. Вивчіть предметну область варіанта завдання і продумайте, яку інформацію слід зберігати про ключовий об'єкті (наприклад, для телефонного тарифу це найменування, вартість, тип і т.д.). Орієнтуючись на Лістинг 3.5.1, створіть невеликий додаток, яке зберігає інформацію про об'єкт обраної предметної області і виводить її на екран.

2. Використовуючи результати попереднього рішення, а також заготовку і Лістинг 4.5.1:

1) усуньте дублювання коду при виведенні списку об'єктів, використовуючи цикли;

2) додайте меню з пунктами:

— список об'єктів;

— фільтр по атрибуту (одному або декільком) об'єкта;

— середній показник по атрибуту об'єкта.

3) перенесіть висновок списку об'єктів до відповідного пункту меню і реалізуйте два нових.

3. Комплексна задача. №5.5.1 . Продовжіть рішення задачі № 4.5.1.

1) Використовуючи результати попереднього рішення, а також заготовку і Лістинг 5.5.1, створіть функції для:

- друку об'єкта;

- «Завантаження» бази об'єктів;

- дій для кожного пункту меню.

2) Для кожної функції:

- продумайте набір вхідних / вихідних параметрів, а також результат і типи значень;

- додайте рядок документації;

- перевірте відсутність побічного ефекту (за винятком функції друку об'єкта).

4. Комплексна задача №9.5.1. Продовжіть рішення задачі № 8.5.1.

Використовуючи результати попереднього рішення, а також заготовку і Лістинги 9.5.1-9.5.3:

— функції збереження і завантаження перенесіть в окремий модуль;

— обчислення і фільтри - в окремий модуль;

— імпортуйте нові модулі, скоректуйте main.py;

— додайте `if __name__ == "__main__"` в main.py.

5. Мережеве та Інтернет-програмування.

— Інформація про книгу

— Статистика ВКонтакте

— Розсилка в міжнародній компанії

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

1. Методичні рекомендації до виконання контрольної роботи

<https://drive.google.com/drive/u/1/folders/1OPFrXAxjUtz2YOiVlUly0QJdida>

H_m0

9 СИСТЕМА ОЦІНЮВАННЯ НАВЧАЛЬНИХ ДОСЯГНЕНЬ **Критерії оцінювання навчальних досягнень студентів**

Модуль № 1 Особливості Python. 40 год

Тиждень початку модуля 1 Тиждень закінчення модуля 5

№ п/п	Критерії	Вага
1.	Постійна обов'язкова частина	
	Лекційна контрольна модульна робота	15
1.2.	Захист лабораторної роботи № 1	15
1.3.	Захист лабораторної роботи № 2	15
	Захист лабораторної роботи № 3	15
	Разом	60
2.	Додаткова варіативна частина	
	Самостійне вивчення нової теми курсу	10
	Систематична робота на лабораторних заняттях	5
	Підготовка реферату	25
	Разом	40
3.	Штрафна частина	
	Несвоєчасний захист лабораторної роботи 1	-10
	Несвоєчасний захист лабораторної роботи 2	-10
	Несвоєчасний захист лабораторної роботи 3	-10
	Незадовільна оцінка лекційної контрольної роботи	-10
	Разом	-40
	ВСЬОГО	100

Модуль № 2 Назва «Підпрограми» 40 год

Тиждень початку модуля 6 Тиждень закінчення модуля 11

№ п/п	Критерії	Вага
4.	Постійна обов'язкова частина	
	Лекційна контрольна модульна робота	15
	Захист лабораторної роботи № 4	15
	Захист лабораторної роботи № 5	15
	Захист лабораторної роботи № 6	15
	Разом	60
5.	Додаткова варіативна частина	
	Самостійне вивчення нової теми курсу	10
	Систематична робота на лабораторних заняттях	20
	Підготовка презентації виконаної роботи	10
	Разом	40
6.	Штрафна частина	
	Несвоєчасний захист лабораторної роботи 4	-10
	Несвоєчасний захист лабораторної роботи 5	-10
	Несвоєчасний захист лабораторної роботи 6	-10
	Незадовільна оцінка лекційної контрольної роботи	-10
	Разом	-40
	ВСЬОГО	100

Модуль № 3 . Мережеве та Інтернет-програмування 40 год

Тиждень початку модуля 12 Тиждень закінчення модуля 17

№ п/п	Критерії	Вага
7.	Постійна обов'язкова частина	
	Лекційна контрольна модульна робота	20
	Захист лабораторної роботи № 7	20
	Захист лабораторної роботи № 8	20

		Разом 60
8.	Додаткова варіативна частина	
	Самостійне вивчення нової теми курсу	10
	Систематична робота на лабораторних заняттях	20
	Підготовка презентації виконаної роботи	10
	Разом	40
9.	Штрафна частина	
	Несвоєчасний захист лабораторної роботи 7	-15
	Несвоєчасний захист лабораторної роботи 8	-15
	Незадовільна оцінка лекційної контрольної роботи	-10
	Разом	-40
	ВСЬОГО	100

9.1 МЕТОДИ КОНТРОЛЮ

Співбесіди, тестування. Форма підсумкового контролю успішності навчання: залік

9.2 ПИТАННЯ ДО ЗАЛІКУ

1. Наведіть основні віхи розвитку мови Python: історію виникнення, ким і коли був створений, цільове призначення, основні версії.

2. Перерахуйте основні характеристики мови програмування Python, вкажіть деякі відмінні риси. Дзен мови Python.

3. Python-програма: лексична і семантична структура, структура файлу, основні елементи і блоки, кодування файлу.

4. Виконання Python-програм: можливі варіанти запуску програм і роботи з інтерпретатором.

5. IDE і їх призначення. Приклади відомих IDE для Python і їх можливості (на будь-якому прикладі).

6. Дайте визначення типу даних і змінної. Наведіть класифікацію вбудованих типів даних в Python. Чим пояснюється наявність великої кількості різних типів?

7. Оператор присвоювання. Управління пам'яттю і збирач сміття.

8. Скалярні типи даних: числа, логічний тип, NoneType. Визначення, основні операції.

9. Послідовності: список, кортеж, числовий діапазон. Визначення, основні операції. Де доцільно використовувати кожен з структур?

10. Безлічі: визначення, основні операції. Де доцільно застосовувати безлічі?
11. Відображення (словник): визначення, основні операції. Де доцільно застосовувати словники?
12. Загальні функції для об'єктів, пріоритет операцій. Чи можна вплинути на пріоритет виконання операцій?
13. Перевірка типів і взаємне перетворення. Для чого може знадобитися перевірка типів і перетворення?
14. Різниця між поверхневою і глибокою копією. Для яких типів глибока копія має сенс і для чого може знадобитися?
15. Введення і виведення в термінал: особливості та приклади.
16. Для чого необхідні конструкції розгалуження і циклів? Чи підтримує їх Python?
17. Умовний оператор if: випадки використання, синтаксис, особливості виконання, приклади.
18. Цикл while: випадки використання, синтаксис, особливості виконання, приклади.
19. Цикл for: випадки використання, синтаксис, особливості виконання, приклади. Ітератори і спеціальні функції.
20. Переривання і продовження циклу: оператори break, continue і ін.
21. Вкладені цикли: необхідність, характерні приклади.
22. Можливість комбінації умовних і циклічних структур. Приклади використання.
23. Колекційні включення: область застосування, приклади.
24. Підпрограма: визначення, правила оголошення. Параметри і аргументи, параметри по посиланню і за значенням. Поняття виклику функції. Стек викликів.
25. Визначення функції в Python, ключове слово return. Правила PEP8 для функцій. Чотири типи функцій в Python.
26. Позиційні і ключові параметри / аргументи функції. Приклади і особливості обчислення.
27. Упаковка і розпакування аргументів.
28. Поняття області видимості. Чотири області видимості в Python. Ключові слова global та nonlocal.
29. Поняття рекурсії і повернення декількох значень з функції. Приклади використання.
30. Рядки документації: необхідність і приклади оформлення.
31. Анонімні функції: доцільність і приклади застосування.
32. Побічні ефекти в програмуванні, основні правила при створенні функцій.
33. Файлова система, файли і каталоги: визначення і термінологія.
34. Властивості файлу. Абсолютний і відносний шлях.
35. Операції над файлами. Види файлів: файли з послідовним і довільним доступом.

36. Файловий об'єкт в Python, функція open (). Обробка винятків при роботі з файловим об'єктом.
37. Основні властивості і методи файлового об'єкта. Приклади читання і запису файлу (одиначне, порядкове, обробка файлу цілком).
38. Серіалізація і десеріалізацію: визначення і області застосування. Модуль pickle.
39. Популярні формати текстових даних. Модуль csv і пакет json. Приклади роботи з CSV і JSON.
40. Модульне програмування. Поняття модуля і пакета, їх реалізація в мові Python.
41. Класифікація модулів / пакетів в Python.
42. Підключення та використання модулів / пакетів. Інструкція import.
43. Область пошуку модулів. Загальна схема імпорту.
44. Спеціальні та додаткові атрибути модуля.
45. Кешування («компіляція») модулів. Особливості * .рус-файлів.
46. Власні модулі: нюанси PEP8, приклади. Різниця між запуском і імпортом модуля.
47. Взаємодія програм по мережі. Приклади. Передача даних, стек протоколів TCP / IP. Структура стека протоколів TCP / IP і загальний хід передачі інформації. Різниця між TCP і UDP.
48. Сокети, IP-адреса і порт. Принципи роботи стека протоколів TCP / IP на прикладі звичайної пошти.
49. Прикладний рівень стека TCP / IP: найбільш відомі протоколи.
50. Інтернет і Веб. Загальні відомості про протоколи HTTP і HTTPS. Поняття URI.
51. Типове взаємодія по протоколу HTTP і його структура. Управління станом і поняття сесії.
52. API веб-сервісів: призначення і відміну від HTML.
53. Електронна пошта: історія і найбільш популярні безкоштовні поштові сервіси.
54. Архітектура і схема відправки електронної пошти.
55. Протоколи відправки / отримання пошти (SMTP, POP3 і IMAP).
56. Формат електронного листа: основні дані конверта і повідомлення.
57. Підтримка стека TCP / IP в Python. Модуль socket.
58. Підтримка HTTP / HTTPS протоколів в Python: бібліотека requests.
59. Робота з електронною поштою в Python: пакет email, модулі smtplib, poplib і imaplib.

9.3 КРИТЕРІЇ ОЦІНЮВАННЯ

ДЕННА ФОРМА

Модуль	Критерії оцінювання знань	Кількість балів
--------	---------------------------	-----------------

	<i>максимум</i>
Змістовий модуль № 1 Особливості Python	
Виконання лабораторної роботи №1	5
Виконання лабораторної роботи №2	5
Виконання лабораторної роботи №3	5
Виконання лабораторної роботи №4	5
<i>Всього за змістовим модулем №1</i>	20
Змістовий модуль № 2 Підпрограми	
Виконання лабораторної роботи №5	10
Виконання лабораторної роботи №6	10
<i>Всього за змістовим модулем №2</i>	20
Змістовий модуль № 3 Мережеве та Інтернет-програмування	
Виконання лабораторної роботи №7	10
Виконання лабораторної роботи №8	10
<i>Всього за змістовим модулем №3</i>	20
Залік	40
Разом	100
<u>Штрафна частина</u>	
Пропуск одного заняття без поважної причини	-5
Здача звіту з лабораторних занять пізніше узгодженого терміну без поважної причини	-10
ЗАЛІК	40
ІТОГОВА СЕМЕСТРОВА ОЦІНКА	100

ЗАОЧНА ФОРМА

Вид навчальної роботи	Кількість балів <i>максимум</i>
Виконання та захист лабораторних робіт №1,2	60
Залік	40
ПІДСУМКОВА СЕМЕСТРОВА ОЦІНКА	100

10 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1-5 джерел

1. Олексій Васильєв Програмування мовою Python / Олексій Васильєв Ceder. -Київ: Навчальна книга - Богдан 2019 - 504 стр.
2. Naomi Ceder The Quick Python Book 3rd Edition / Naomi Ceder. - NY: Manning Publications Co., 2018 - 432 p.

3. Kenneth A. Lambert Fundamentals of Python: first programs / Kenneth A. Lambert. NY: Cengage Learning, 2018 - 476 p.
4. Mark Lutz Programming Python / Mark Lutz. - NY: O'Reilly Media, Inc., 2011 - 1628 p.
5. Николай Прохоренок Python 3 и PyQt 5. Разработка приложений / Прохоренок Н.А., Дронов В.А. - СПб.: БХВ-Петербург, 2016. - 832 с.

Допоміжна

5-10 джерел

1. С. Шапошникова. Основы программирования на Python. Учебник. Вводный курс. — версия 2. — 2011. — 44 с.
2. И. А. Хахаев. Практикум по алгоритмизации и программированию на Python. Учебник —М.: Альт Линукс, 2010. — 126 с.
3. А. Н. Чаплыгин. Учимся программировать вместе с Питоном. Учебник, ревизия 226. — 135 с.
4. Г. Россум, Ф. Л. Дж. Дрейк, Д. С. Откидач и др. Язык программирования Python (не издано, черновик). — СПб.: АНО «Институт логики». Невский Диалект, 2001. — 454 с.
5. A. Svveigart. Invent Your Own Computer Games with Python. — 2008—2010. 436 с.
6. A. Svveigart. Core Python Applications Programming. — 2012. — 888 с.

11 ІНФОРМАЦІЙНІ РЕСУРСИ

1. Топ-5 книг, які повинен прочитати кожен розробник ПЗ. URL: <https://habrahabr.ru/post/135897/>;
2. Школа системного аналізу: Книги. URL: [http://systems.education/books](http://systems.education/books;);
3. Які книги треба прочитати розробнику: Обговорення в Facebook. URL: <https://appttractor.ru/info/articles/kakie-knigi-nado-prochitat-razrabotchiku-obsuzhdenie-v-facebook.html>;
4. 7 книг, які варто прочитати для вивчення Python. URL: <https://proglib.io/p/7-python-books/>;
5. Ідеальна добірка книг для розробників. URL: <https://proglib.io/p/ultimate-reading-list/>;
6. Reading List: Fog Creek Software Management Training Program. URL: <https://www.joelonsoftware.com/2005/11/22/reading-list-fog-creek-software-management-training-program/>.
7. Офіційний сайт Python. URL: <https://www.python.org/>.
8. Документація Python. URL: <https://docs.python.org/3/>.
9. Python Software Foundation. URL: https://ru.wikipedia.org/wiki/Python_Software_Foundation.
10. PyPI - the Python Package Index. URL: <https://pypi.python.org/pypi>.

11. Full Grammar specification. URL: <https://docs.python.org/3/reference/grammar.html>.
12. Should I use Python 2 or Python 3 for my development activity ?. URL: <https://wiki.python.org/moin/Python2orPython3>.
13. Python2 vs Python3: відмінності синтаксису. URL: <http://pythonworld.ru/osnovy/python2-vs-python3-razlichiya-sintaksisa.html>.
14. Установка і використання Python (поглиблений опис). URL: <https://docs.python.org/3/using/index.html>.
15. Перше повідомлення про Python. URL: <http://svn.python.org/view/checkout/python/trunk/Misc/HISTORY>.
16. Guido van Rossum on the History of Python. URL: <https://www.youtube.com/watch?v=ugqu10JV7dk>.
17. Guido van Rossum: The Modern Era of Python. URL: <https://youtu.be/rTTFh7HOIC0>.
18. PEP 404 - Python 2.8 Un-release Schedule (чому Python 2.8 ніколи не буде). URL: <https://www.python.org/dev/peps/pep-0404/>.
19. Персона. Гвідо Ван Россум - творець Python. URL: <https://habrahabr.ru/post/315974/>.
20. Ofitsiynyy sayt Python. URL: <https://www.python.org/>.

12 ПЕРЕЛІК НОРМАТИВНИХ ДОКУМЕНТІВ

за необхідністю

1. Офіційний сайт Python. URL: <https://www.python.org/>.
2. Документація Python. URL: <https://docs.python.org/3/>.

13 ПОЛІТИКА ДИСЦИПЛІНИ

Форми організації освітнього процесу, види навчальних занять і оцінювання результатів навчання регламентуються Положенням про організацію освітнього процесу.

Політика виставлення оцінок: кожна оцінка виставляється відповідно до розроблених викладачем та заздалегідь оголошених студентам критеріїв, а також мотивується в індивідуальному порядку; у випадку не виконання студентом усіх передбачених навчальним планом видів заняття (лабораторних робіт) до заліку він не допускається; пропущені заняття обов'язково мають бути відпрацьовані.

Відвідування є обов'язковим (за винятком випадків, коли існує поважна причина, наприклад, хвороба чи дозвіл працівників деканату). Якщо студент не може бути присутнім на заняттях, він все одно несе відповідальність за виконання завдань, що проводились в комп'ютерному класі.

Порядок зарахування пропущених занять. Відпрацювання пропущеного заняття з лекційного курсу здійснюється шляхом підготовки і захисту реферату за відповідною темою у вигляді презентації. Захист реферату відбувається відповідно до графіку консультацій викладача, з яким можна

ознайомитись на кафедрі. Відпрацювання пропущеного лабораторного заняття здійснюється шляхом самостійного виконання завдання і його захисту відповідно до графіку консультацій викладача.

Політика академічної поведінки та доброчесності: конфліктні ситуації мають відкрито обговорюватись в академічних групах з викладачем, необхідно бути взаємно толерантним, поважати думку іншого. Плагіат та інші форми нечесної роботи неприпустимі. Всі індивідуальні завдання студент має виконати самостійно із використанням рекомендованої літератури й отриманих знань та навичок. Цитування в письмових роботах допускається тільки із відповідним посиланням на авторський текст. Недопустимі підказки і списування у ході захисту лабораторних робіт, на контрольних роботах, на заліку.

Норми академічної етики: дисциплінованість; дотримання субординації; чесність; відповідальність; робота в аудиторії з відключеними мобільними телефонами. Повага один до одного дає можливість ефективніше досягати поставлених командних результатів. При виконанні лабораторних робіт студент може користуватися ноутбуками. Проте під час лекційних занять та обговорення завдань лабораторних робіт не слід використовувати ноутбуки, смартфони, планшета чи комп'ютери. Це відволікає викладача і студентів групи та перешкоджає навчальному процесу. Якщо ви використовуєте свій ноутбук чи телефон для аудіо- чи відеозапису, необхідно заздалегідь отримати дозвіл викладача.