



Course of study (code) / Назва дисципліни (шифр)	Основи програмування на Python IT1211BOPNP	
Academic year / Навчальний рік - Семестр	2022/2023 –2 семестр	
Course of study / Назва спеціальності	121 Інженерія програмного забезпечення	
Educational program / Освітня програма	«Інженерія програмного забезпечення» Перший (бакалаврський) рівень -4 ECTS	
Education - ECTS / Рівень – Кредити Status / Статус	Вибіркова Українська	
Learning language / Мова навчання		
Author / Укладач	Дмитро Медведєв, кандидат технічних наук, Державний університет економіки і технологій, e-mail: medvediev_dq@kneu.dp.ua, http://orcid.org/0000-0002-3747-1717 моб. +380688535681	
Консультації	чт, 14.00-15.00	

A. OBJECTIVE OF THE SUBJECT / МЕТА ТА ЗАВДАННЯ ДИСЦИПЛІНИ

Мета викладання навчальної дисципліни - це забезпечити отримання студентами теоретичних знань і практичних навичок прикладного програмування. За результатами вивчення дисципліни студент буде знати базові алгоритмічні конструкції та їх представлення на мові Python, засоби представлення в програмі даних та базових структур даних, формує вміння та навички складання алгоритмів різної складності, буде вміти оцінювати їх ефективність, виконувати тестування та налагодження. Також за підсумками вивчення дисципліни студент бути мати досвід використовувати отримані знання та вміння на практиці при розв'язуванні прикладних та нестандартних завдань.

B. SUBJECT PROGRAM / ПРОГРАМА ДИСЦИПЛІНИ

Тема 1. Огляд мови Python.

Чому використовують Python? Де використовують Python? Які проекти зроблено на Python? Переваги. Недоліки. Трансляція коду на Python. Виконання коду з точки зору програміста. Виконання коду з точки зору Python. Різновиди Python. Командний рядок. Середовища розробки коду

Тема 2. Типи даних,.

Числа. Рядки. Списки. Словники. Кортежі. Файли. Динамічна типізація

Тема 3. Оператори.

Оператор присвоєння. Арифметичні оператори. Побітові оператори. Оператори порівнянь. Логічні оператори. Пріоритет операторів. Оператори вибору. Оператори циклу. Оператори переходу

Тема 4. Оператор вибору варіанта дії на основі результатів перевірки.

Оператор if. Загальний формат. Багаторівневість. Обмеження блоків: правила відступів. Обмеження операторів: рядки і їх переноси. Значення істинності. Булеві перевірки. Тернарний вираз. Відслідковування помилок

Тема 5. Циклічні оператори.

Цикли while. Загальний формат. Оператори break, continue, pass, else. Цикл for. Загальний формат. Метод range. Порівняння комбінації while і range з for. Методи len, zip, map, enumerate

Тема 6. Функції.

Оператор def. Визначення та виклик функції. Повернення результату. Поліморфізм в Python. Області видимості: локальні змінні, оператори global та local. Аргументи функції: фіксована кількість, динамічна кількість, ключові значення.

Тема 7. Розширені можливості функцій.

Концепції проектування функцій. Рекурсивні функції. Стек. Порівняння рекурсії з циклом. Об'єкти функцій. Атрибути функцій. Анонімні функції. Генератори

Тема 8. Модулі та пакети.

Призначення модулів. Структурування програми. Імпортування та атрибути. Стандартні бібліотечні модулі. Пошук файлу модуля. Компіляція файлу модуля. Виконання модуля.

C. LIST OF COMPETENCIES AND STUDIES TARGETED RESULTS / ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ ТА ПРОГРАМНИХ РЕЗУЛЬТАТІВ НАВЧАННЯ

Загальні компетентності (ЗК)	ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу. ЗК-2. Здатність застосовувати знання у практичних ситуаціях. ЗК-3. Здатність проведення теоретичних та прикладних досліджень на відповідному рівні ЗК-8. Знати класифікацію програмного забезпечення та призначення його складових частин. ЗК-9. Здатність використовувати сучасні інтегровані прикладні програмні системи для обробки числової, текстової та графічної інформації.
Спеціальні (фахові) компетентності (ФК)	СК 2. Здатність розробляти архітектури, модулі та компоненти програмних систем. СК 3. Знання і розуміння специфікацій, стандартів, правил і рекомендацій в професійній галузі, уміння оцінювати ступінь обґрунтованості їх застосування, здатність дотримуватися їх при реалізації процесів життєвого циклу.



	<p>СК 4. Здатність забезпечувати технічну підтримку і навчання користувачів програмного забезпечення.</p> <p>СК 5. Уміння готувати та презентувати документацію та методичні матеріали щодо програмного забезпечення.</p> <p>СК 6. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.</p> <p>СК 7 Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності загальної функціональності і надійності програмного забезпечення.</p> <p>СК 8. Здатність до алгоритмічного та логічного мислення.</p> <p>СК 9. Здатність приймати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p>
Програмні результати навчання (ПРН)	<p>ПРН 3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПРН 4. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПРН 7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПРН 13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p>

D. SEMESTER PLAN / СЕМЕСТРОВИЙ ПЛАН

Тиждень/ Дата	Тема, план/короткі тези	Форма діяльності (заняття), години, формат	Завдання для СРС (література, ресурси в інтернеті, презентація, відеокурси)
1 тиждень	<p><i>Тема 1. Огляд мови Python.</i> Чому використовують Python? Де використовують Python? Які проекти зроблено на Python? Переваги. Недоліки. Трансляція коду на Python. Виконання коду з точки зору програміста. Виконання коду з точки зору Python. Різновиди Python. Командний рядок. Середовища розробки коду</p>	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 3, 5 додаткова 3
		Лабораторні заняття (2 год), F2F	Переваги. Недоліки. Трансляція коду на Python. Виконання коду з точки зору програміста. Виконання коду з точки зору
2-3 тиждень	<p><i>Тема 2. Типи даних, .</i> Числа. Рядки. Списки. Словники. Кортежі. Файли. Динамічна типізація.</p>	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 3, 5 додаткова 3
		Лабораторні заняття (4 год), F2F	Числа. Рядки. Списки. Словники. Кортежі. Файли. Лабораторна робота №2
4-5 тиждень	<p><i>Тема 3. Оператори.</i> Оператор присвоєння. Арифметичні оператори. Побітові оператори. Оператори порівнянь. Логічні оператори. Пріоритет операторів. Оператори вибору. Оператори циклу. Оператори переходу</p>	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3, 5, 8 додаткова 1,2,3
		Лабораторні заняття (4 год), F2F	Арифметичні оператори. Побітові оператори. Оператори порівнянь. Логічні оператори. Пріоритет операторів. Оператори вибору Лабораторна робота №3
6-7 тиждень	<p><i>Тема 4. Оператор вибору варіанта дії на основі результатів перевірки.</i> Оператор if. Загальний формат. Багаторівневість. Обмеження блоків: правила</p>	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3, 5, 7, 8 додаткова 1, 2, 3
		Лабораторні заняття (4 год),	Оператор if. Загальний формат. Багаторівневість.



	відступів. Обмеження операторів: рядки і їх переноси. Значення істинності. Булеві перевірки. Тернарний вираз. Відслідковування помилок	F2F	Тернарний вираз. Відслідковування помилок Лабораторна робота №4
8-11 тиждень	Тема 5. <i>Циклічні оператори</i> . Цикли while. Загальний формат. Оператори break, continue, pass, else. Цикл for. Загальний формат. Метод range. Порівняння комбінації while і range з for. Методи len, zip, map, enumerate	Лекція, (4 год), F2F	Опрацювання літератури: основна 1, 5, 7, 8 додаткова 1,2,3,4
		Лабораторні заняття (4 год), F2F	Цикли while. Загальний формат. Оператори break, continue, pass, else. Цикл for. Лабораторна робота №5
12-13 тиждень	Тема 6. <i>Функції</i> . Оператор def. Визначення та виклик функції. Повернення результату. Поліморфізм в Python. Області видимості: локальні змінні, оператори global та local. Аргументи функції: фіксована кількість, динамічна кількість, ключові значення.	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (4 год), F2F	Визначення та виклик функції. Повернення результату. Поліморфізм в Python. Області видимості Лабораторна робота №6
14-15 тиждень	Тема 7. <i>Розширені можливості функцій</i> . Концепції проектування функцій. Рекурсивні функції. Стек. Порівняння рекурсії з циклом. Об'єкти функцій. Атрибути функцій. Анонімні функції. Генератори	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (4 год), F2F	Рекурсивні функції. Порівняння рекурсії з циклом. Об'єкти функцій. Атрибути функцій. Анонімні функції. Лабораторна робота №7
16-17 тиждень	Тема 8. <i>Модулі та пакети</i> . Призначення модулів. Структурування програми. Імпортування та атрибути. Стандартні бібліотечні модулі. Пошук файлу модуля. Компіляція файлу модуля. Виконання модуля.	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (4 год), F2F	Призначення модулів. Стандартні бібліотечні модулі. Пошук файлу модуля. Виконання модуля Лабораторна робота №8
18 тиждень	Заняття для захисту індивідуальних робіт.	Лабораторне заняття (2 год), F2F	Виправлення недоліків індивідуальних робіт, виконання програмного коду, розрахунок контрольного прикладу

Вивчення дисципліни передбачає виконання двох аудиторних тестових завдань за допомогою програми Zelis. Перша контрольна модульна робота виконується за темами 1-4, друга – за темами 5-8 у тестовій формі. Під час виконання студенти мають продемонструвати уміння та навички залучати набуті теоретичні та практичні знання з програмування на мові Python.

Замість науково-дослідницької роботи студенти можуть отримати 10 балів за вивчення онлайн курсу «Програмування на мові Python» на платформі COURSERA (<https://www.coursera.org/>), посилання на курс: <https://www.coursera.org/specializations/python>

Детальний план проведення лабораторних занять, завдання для індивідуальних робіт містяться в системі MOODLE у відповідних розділах.

E. BASIC LITERATURE (OBLIGATORY TEXTBOOKS) / ОСНОВНА ЛІТЕРАТУРА (ОБОВ'ЯЗКОВІ ПІДРУЧНИКИ)

1. Путівник мовою програмування Python [Електронний ресурс]. – Режим доступу : <http://pythonguide.rozh2sch.org.ua>.
2. ArcGIS Pro Python [Електронний ресурс]. – Режим доступу : <https://pro.arcgis.com/ru/pro-app/arcpy/main/arcgis-pro-arcpy-reference.htm>.
3. Python. Обучение программированию [Электронный ресурс]. – Режим доступа : <https://younglinux.info/python>
4. Python ООП: <https://python-scripts.com/object-oriented-programming-in-python#pros-cons-oop>
5. Лутц М. Изучаем Python / М. Лутц. – СПб. : Символ-Плюс, 2011. – 1280 с.
6. Лутц М. Програмування на Python : в 2 томах / М. Лутц. – СПб. : Символ-Плюс, 2011. – Т. 1. – 992 с.



7. Дэвид М. Бизли Python. Подробный справочник / Дэвид М. Бизли. – СПб. : Символ-Плюс, 2010. – 864 с.
8. Саммерфилд М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. – СПб. : Символ-Плюс, 2009. – 608 с.
9. Саммерфилд М. Python на практике / М. Саммерфилд – М. : ДМК Пресс, 2014. – 338 с.
10. Сузи Р. А. Язык программирования Python : учеб. пособие / Р. А. Сузи. – М. : ИНТУИТ, БИНОМ. Лаборатория знаний, 2006. – 328 с.
11. Доусон М. Програмуємо на Python / М. Доусон. – СПб. : Питер, 2012. – 432 с.
12. Хахаев И. А. Практикум по алгоритмизации и программированию на Python: учебник / И. А. Хахаев. – М. : Альт Линукс, 2010. – 126 с.

F. COMPLEMENTARY LITERATURE / ДОДАТКОВА ЛІТЕРАТУРА

1. А. Мюллер, С. Гвидо - Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными – М. 2017. — 393 с.
2. Буйначев С.К., Боклаг Н.Ю. - Основы программирования на языке Python – Екатеринбург. - 2014. — 90 с.
3. Бэрри П. - Изучаем программирование на Python (Мировой компьютерный бестселлер). - М.- 2017. — 618 с.
4. Зед Шоу - Легкий способ выучить Python (Мировой компьютерный бестселлер) - 2017. — 353 с.
5. Федоров Д. - Основы программирования на примере языка Python - М.- 2018. – 167 с.
Чан Уэсли Дж. - Python. создание приложений (Библиотека профессионала) –М. – 2015.- 794 с.

G. THE MOST IMPORTANT PUBLICATIONS OF THE AUTHOR(S) CONCERNING PROPOSED CLASSES / ОСНОВНІ ПУБЛІКАЦІЇ АВТОРА, ЩО ПОВ'ЯЗАНІ З ТЕМАТИКОЮ ЗАПЛАНОВАНИХ ЗАНЯТЬ

- 1.

H. PREREQUISITE AND POSTREQUISITE / ПРЕРЕКВІЗИТИ ТА ПОСТРЕКВІЗИТИ

I. SCOPE AND TYPE / КІЛЬКІСТЬ ВІДВЕДЕНИХ ГОДИН ТА ФОРМА ПРОВЕДЕННЯ ЗАНЯТЬ

	Денна	Заочна
Лекції	18	6
Практичні (лабораторні)	36	6
Самостійна робота студента (СРС)	54	96
Індивідуально-консультативна робота (ІКР)	12	12
Курсова робота	–	–

J. CURRENT AND FINAL EVALUATION / ПОТОЧНЕ ТА ПІДСУМКОВЕ ОЦІНЮВАННЯ

	Денна	Заочна
Поточний контроль, в т.ч.:	50	50
оцінювання під час аудиторних занять	10	10
виконання контрольних (тестових) робіт	10	10
виконання і захист завдань самостійної роботи	20	20
науково-дослідницька робота	10	10
Підсумковий контроль (екзамен)	50	50
Разом	100	100

J. CURRENT AND FINAL EVALUATION / ПОТОЧНЕ ТА ПІДСУМКОВЕ ОЦІНЮВАННЯ

	Денна	Заочна
Шкала балів	Оцінка за 4-бальною шкалою	Шкала ECTS
90 – 100	Відмінно	A
80 – 89	Добре	B
70 – 79		C
66 – 69		D
60 – 65	Задовільно	E
21 – 59	незадовільно з можливістю повторного складання екзамену (заліку)	FX
0 – 20	незадовільно з можливістю вивчення дисципліни за індивідуальним графіком у формі додаткової індивідуально-консультативної роботи.	F

K. CODE OF CONDUCT OF THE COURSE / КОДЕКС ПОВЕДІНКИ ПІД ЧАС ВИВЧЕННЯ КУРСУ

Для успішного проходження курсу та складання контрольних заходів необхідним є виконання наступних обов'язків:

- не запізнюватися на заняття;



- не пропускати заняття (як лекційні, так і практичні), в разі хвороби мати довідку або її ксерокопію;
- самостійно опрацьовувати весь лекційний матеріал та ресурси для самостійної роботи;
- конструктивно підтримувати зворотній зв'язок з викладачем на всіх етапах проходження курсу (особливо під час виконання індивідуальних проектів/курсового проекту);
- своєчасно і самостійно виконувати всі передбачені програмою лабораторні завдання;
- брати очну участь у контрольних заходах;
- будь-яке відтворення результатів чужої праці (включаючи практичну роботу над командним проектом), в тому числі використання завантажених з Інтернету матеріалів, як власних результатів, кваліфікується, як порушення норм і правил академічної доброчесності, та передбачає притягнення до відповідальності у порядку, визначеному чинним законодавством.

L. METHODS OF CONDUCTING / МЕТОДИ НАВЧАННЯ

Для формувань компетентностей застосовуються такі методи навчання:

вербальні/словесні (лекція, пояснення, дискусія);

наочні (спостереження, ілюстрація, демонстрація, відео-уроки);

практичні (різні види лабораторних завдань, вирішення задач, проведення експерименту, практики);

пояснювально-ілюстративний, який передбачає пред'явлення готової інформації викладачем та її засвоєння студентами;

метод проблемного викладу;

дослідницький.

M. TOOLS, EQUIPMENT AND SOFTWARE / ІНСТРУМЕНТИ, ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

мультимедійний клас з ПК, цифровий проектор, інтегрована середа IDLE Python

[Zoom](#) – хмарна платформа для відео і аудіо конференцій та вебінарів

Тестування знань та отримання оцінки з екзамену:

ZELIS - система призначена для тестування знань студентів в двох режимах: автоматизований контроль знань та тестування по бланкам.

N. STUDENT RESOURCES, MOOC PLATFORMS / ЦИФРОВІ РЕСУРСИ ДЛЯ СТУДЕНТІВ ТА ВІДКРИТІ ДИСТАНЦІЙНІ ОНЛАЙН КУРСИ

Студентам пропонується доступ до навчальних матеріалів дисципліни - <http://moodle.kneu.dp.ua>.

[Coursera](#) – безкоштовні онлайн-курси з різних дисциплін, у разі успішного закінчення яких користувач отримує сертифікат про проходження курсу.

[EdX](#) – онлайн-курси від закладів вищої освіти.

[Prometheus](#) – український громадський проект масових відкритих онлайн-курсів.

O. FEEDBACK/ ЗВОРОТНІЙ ЗВ'ЯЗОК

Електронні листи є найкращим способом зв'язатися з керівником курсу, і, будь ласка, додайте шифр групи в темі листа. Якщо ви надішлете мені електронне повідомлення, надайте мені, принаймні, 48 годин (протягом робочого тижня), щоб відповісти. Якщо ви не отримаєте відповідь, відправте листа повторно.

P. ACADEMIC HONESTY/ АКАДЕМІЧНА ДОБРОЧЕСНІСТЬ

Державний університет економіки і технологій очікує від студентів розуміння та підтримання високих стандартів академічної чесності. Приклади академічної не доброчесності включають такі: плагіат, зловживання інформацією із застарілих джерел мережі. Очікується, що вся робота, виконана відповідно до вимог курсу, є власною роботою студента. Під час підготовки роботи, яка відповідає вимогам курсу, студенти повинні відрізняти власні ідеї від інформації, отриманої з інших джерел. Без попереднього письмового схвалення викладачем, студенти можуть не подавати один і той же звіт двічі.

APPROVED / ЗАТВЕРДЖЕНО

Рішенням кафедри інформатики і ППЗ Державного університету економіки і технологій - протокол №__ від __.__.2022 року

Укладач:

Дмитро МЕДВЕДСВ

ЗАТВЕРДЖЕНО:

Кафедрою інформатики і прикладного програмного забезпечення

Протокол № 1 від 25 серпня 2022 року

В.о. завідувача кафедри

Олександр ЗЕЛЕНСЬКИЙ

Науково-методичною радою Державного університету економіки і технологій

Протокол № 1 від 20 вересня 2022 року

Голова науково-методичної ради

Валентин ОРЛОВ