**Кафедра кібербезпеки та математичного моделювання**

|  |  |
| --- | --- |
| **Назва курсу** | Технології програмування |
| **Мова викладання** | українська |
| **Курс та семестр вивчення** | 2 курс, 3семестр |
| **Викладач** | Дюба Ігор Миколайович, асистент |
| **Профайл викладача** | Сайт кафедри: <https://mmi.stu.cn.ua/personal-kafedry/> |
| **Контакти викладача** | тел.: 063 837 56 34  e-mail: idyuba@gmail.com |

**1.** **Анотація курсу** - навчальна дисципліна «Технології програмування» є обов’язковою дисципліною освітньої програми «Кібербезпека Вивчення дисципліни дозволяє здобувачам вищої освіти оволодіти базовими принципами для вивчення мови програмування, що можуть бути використанні в майбутньому при необхідності переходу на нову мову програмування, досвід використання якої відсутній.

Під час вивчення дисципліни здобувачі вищої освіти вивчають принципи використання можливостей мови програмування для побудови алгоритмів вирішення завдань. Основною метою дисципліни є отримання практичного досвіду використання нової мови програмування.

Мова програмування Python вибрана завдяки своїй легкості використання та швидкому розгортанню. Python – це універсальна мова, що широко використовується в усьому світі для самих різних цілей. Серед переваг мови Python можна виділити переносимість написаних програм, на комп’ютери різної архітектури та з різними операційними системами, лаконічність запису алгоритмів, можливість отримати ефективний код програм за швидкістю виконання.

Посилання на курс в MOODLE: <https://eln.stu.cn.ua/course/view.php?id=7199>

**2. Мета та цілі курсу -** формування науково-професійного світогляду бакалавра спеціальності 125 – Кібербезпека та захист інформації в області сучасних парадигм програмування під час програмної реалізації завдань. В процесі слухання курсу студенти ознайомляться з принципами здобуття теоретичних знань в галузі нової мови програмування, застосування теоретичних знання на практиці, проведення поверхневого аналізу завдання та розбивку на під задачі, принципами вибору між існуючими варіантами для вирішення завдання, перевірки результатів роботи.

Об’єкт – підходи вивчення нової мови програмування.

Предметом вивчення навчальної дисципліни є мова програмування Python.

Під час вивчення дисципліни здобувач вищої освіти має набути або розширити наступні загальні та фахові компетентності, передбачені освітньою програмою спеціальності 125 – Кібербезпека та захист інформації:

КЗ1.Здатність застосовувати знання у практичних ситуаціях.

КЗ2. Знання та розуміння предметної області та розуміння професії.

КЗ4. Вміння виявляти, ставити та вирішувати проблеми за професійним спрямуванням

Основними завданнями навчальної дисципліни «Технології програмування» є:

* сформувати у здобувачів вищої освіти знання про базові елементи та конструкції мов програмування, синтаксичні особливості, методи обробки виняткових ситуацій, методи роботи з файлами та базові принципи тестування коду;
* вміти програмувати функціонально;
* вміти оцінювати правильність, структуру та стиль коду;
* сформувати основні поняття щодо тестування коду та знаходження помилок;
* вміти використовувати технічну документацію для виконання поставлених завдань;
* підготувати студентів для використання отриманих знань при необхідності вивчення інших мов програмування.

Серед головних цілей цього курсу отримання практичного досвіду студентами в опанування нової мови програмування, виділення типових підходів для опанування нової мови програмування та використання досвіду для вивчення нових мов програмування в майбутньому.

**3. Результати навчання:**

Під час вивчення дисципліни здобувач вищої освіти має досягти або вдосконалити наступні програмні результати навчання, передбачені освітньою програмою**:**

ПРН 15. Використовувати сучасне програмно-апаратне забезпечення інформаційно-комунікаційних технологій;

У результаті вивчення навчальної дисципліни здобувач вищої освіти повинен знати:

* базові елементи та конструкції мов програмування;
* основний синтаксис мов програмування Python;
* принципи обробки виняткових ситуацій;
* базові принципи тестування написаного коду;
* базові принципи роботи з файлами;
* базові принципи об’єктно орієнтованого підходу;

У результаті вивчення навчальної дисципліни здобувач вищої освіти повинен вміти:

* програмувати функціонально;
* представляти та обробляти інформацію;
* оцінювати правильність, структуру та стиль коду;
* вивчати нові мови програмування;
* читати документацію, робити висновки із специфікацій;
* тестувати програми , знаходити помилки та виявляти проблемні випадки;
* точно описувати симптоми помилок і чітко задавати питання.

**4. Обсяг курсу.** 4 кредити ECTS, що становить 120 годин роботи студентів, з них 90 годин самостійної роботи + 30 годин аудиторної роботи з викладачем.

|  |  |
| --- | --- |
| Вид заняття | Загальна к-сть годин |
| Лекції | 16 годин |
| Лабораторні заняття | 14 годин |
| Самостійна робота (реферати, презентації, доповіді, КР, наукові дослідження, тощо) | 90 годин |

**Тематика курсу**

**Тема 0. Від ідеї до імплементації**

Все починається з ідеї. Процес перетворення команд з людської мови в інструкції для обчислювальної системи. Підготовчі дії перед початком роботи: оцінка існуючих засобів, розробка архітектури, визначення набору інструментів. Вибір мови програмування, що відповідає критеріям. Основні принципи роботи інтерпретатора. Набір інструментів для роботи: інтерпретатор, середа розробки, git клієнт.

**Тема 1. Функції та змінні**

Вітаю всесвіт, як початок вивчення мові програмування. Набір базових функцій під кожну мову програмування. Аргументи для специфікації виконання функції. Можливі помилки при використанні функцій. Необмежене використання функцій. Коментарі. Результат роботи функції. Поняття змінної. Типи даних. Рядки. Числа. Булеві значення. Написання власних функцій.

**Тема 2. Умовні переходи**

Виконання інструкцій в залежності від результату відповіді на питання. Порівняння як основа запитання. Ключове слово if. Ключове слово else. Ключове слово elif. Компонування декількох питань. Ключове слово or. Ключове слово end. Змінні типу bool.

**Тема 3. Цикли**

Можливість виконувати одні і ті самі дії певну кількість раз. Ключове слово while. Ключове слово for. Складна структура даних список. Довжина списку. Основні функції для роботи зі списком. Складна структура даних словник. Комбіноване використання списків та словників. Складна структура даних кортеж. Відмінність між структурами даних.

**Тема 4. Виняткові ситуації**

Дії, що призводять до появи виняткових ситуацій. Синтаксичні помилки, що виникають в процесі написання коду. Помилки реального часу виконання. Ключове слово try. Ключове слово except. Exception як узагальнення всіх помилок. Ключове слово else для визначення інструкцій безпомилкового виконання. Ключове слово pass. Ключове слово raise.

**Тема 5. Бібліотеки**

Бібліотека як набір файлів з кодом для використовувати у розробці. Сторонні та власні бібліотеки. Модуль як механізм забезпечення повторного використання коду. Ключове слово import. Ключове слово from. Пакет як варіант сторонньої бібліотеки. Веб ресурс PyPI. Встановлення пакетів реалізовано за допомогою скрипту pip.

**Тема 6. Робота з файлами**

Механізм довготривалого зберігання та використання інформації. Функція open(). Функція close(). Функції read() та write(). Ключове слово with. Файли в форматі CSV та однойменний модуль. Основні методи роботи з CSV файлами.

**Тема 7. Об’єктно орієнтовне програмування**

Процедурний підхід написання коду. ООП – інша парадигма вирішення завдань в програмуванні. Використання абстракцій як підхід в написання коду. Можливість створення нових типів даних з вказанням імені. Ключове слово class для опису власних типів даних, що є основною характеристикою ООП. Побудова абстракція object. Атрибути класу для опису властивостей абстракції object. Можливість задавати поведінку абстракції використовуючи підходи поняття methods. Розробка власної поведінки для абстракції. Можливість посиленого контролю за атрибутами абстракції використовуючи підходи поняття properties.

**5.** **Пререквізити та постреквізити:** Дисципліна є складовою частиною фундаментальної підготовки та відноситься до навчальних дисциплін циклу «Обов’язкові дисципліни професійної підготовки» за спеціальністю «Кібербезпека та захист інформації» (бакалавр). Вивчення курсу передбачає наявність систематичних та ґрунтовних знань із курсу – «Інформатика».

Дисципліна є базовою для подальшого використання під час проходження різних видів практик.

**6. Система оцінювання та вимоги**

|  |  |
| --- | --- |
| **Загальна система оцінювання курсу** | |
| **Лабораторні роботи** | Кожна виконана лабораторна робота оцінюється від 0 до 10-ти балів. Кількість балів залежить від рівня теоретичних знань та практичних навичок студента за темою, самостійності виконання роботи та вчасності її захисту |
| **Розрахунково-графічна робота** | При перевірці та оцінюванні РГР враховується правильність виконання теоретичних та практичних завдань, самостійність виконання, вчасність здачі роботи та відповідність оформлення результатів діючим вимогам. Максимальна кількість балів - 35. |
| **Умови допуску до підсумкового контролю** | Умовою допуску до екзамену є виконання та отримання хоча б мінімальної кількості балів з усіх обов'язкових видів навчальної роботи передбачених робочою програмою (лабораторних, модульного контролю та РГР). Мінімальна кількість балів необхідна для допуску до екзамену - 35. |

|  |  |  |
| --- | --- | --- |
| **Діяльність ЗВО та форма контролю** | **Кількість балів** | |
| Якість виконання лабораторних робіт. Рівень знань студента за темою лабораторної роботи (максимум - 5 бал за кожну лаб. роботу) | 0 | 20 |
| Самостійність та своєчасність виконання лабораторних робіт (максимум - 5 бали за кожну лаб. роботу) | 0 | 20 |
| Оцінка за розрахунково-графічну роботу | 0 | 35 |
| Семестрова оцінка поточного контролю | 0 | 75 |
| Екзамен | 0 | 25 |

**7.** **Політики курсу**

Вивчення курсу базується на принципах взаємної поваги й довіри викладача та здобувача вищої освіти, рівноправності й толерантності.

**7.1 Академічна доброчесність** - самостійність виконання навчальних завдань та посилання на джерела у випадку використання напрацювань інших авторів. Види порушень академічної доброчесності - академічний плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво.

Відповідно до [Кодексу академічної доброчесності Національного університету «Чернігівська політехніка»](https://stu.cn.ua/wp-content/stu-media/normobaza/normdoc/norm-yakist/kodeks-akademichnoyi-dobrochesnosti.pdf) за порушення академічної доброчесності здобувачі освіти можуть мати наслідком: повторне проходження підсумкового оцінювання; повторне вивчення відповідного освітнього компонента освітньої програми; - відрахування з Університету; позбавлення академічної стипендії.

**7.2 Політика дедлайнів** - своєчасність здачі лабораторної роботи оцінюється в 1 бал за кожну лабораторну роботу. Своєчасність здачі розрахунково-графічної роботу оцінюється в 2 бали. Відповідно, максимальна оцінка за невчасно здані роботи зменшується на зазначену кількість балів. Виключенням може бути наявність поважних причин несвоєчасної здачі зазначених робіт (хвороба, відсутність електроенергії чи зв’язку, участь в зазначений час в інших видах навчальної, наукової чи організаційної роботи, офіційна робота за фахом).

**7.3 Політика перезарахування кредитів у випадку мобільності** - Перезарахування відбувається якщо назви навчальних дисциплін ідентичні або мають незначну стилістичну відмінність, але обсяги та змістова частина навчальних програм не відрізняються; кількість кредитів, відведена на вивчення навчальної дисципліни відрізняється менше, ніж на 25 %; форми підсумкового контролю з дисциплін однакові. При перезарахуванні дисципліни зберігається раніше здобута позитивна оцінка. Перескладання іспиту з дисципліни з метою підвищення оцінки, визначеної в документах виданих здобувачу вищої освіти за попереднім місцем навчання, не дозволяється.

Перезарахування кредитів проводиться відповідно до [Порядку визначення академічної різниці та визнання результатів попереднього навчання в Національному університеті «Чернігівська політехніка»](https://stu.cn.ua/wp-content/stu-media/normobaza/normdoc/norm-osvitproces/poryadok-vyznachennya-akademichnoi-riznyczi-ta-vyznannya-rezultativ-poperednogo-navchannya.pdf) та [Положення про академічну мобільність учасників освітнього процесу Національного університету «Чернігівська політехніка»](https://stu.cn.ua/wp-content/stu-media/normobaza/normdoc/norm-osvitproces/polozhennya-pro-akademichnu-mobilnist-uchasnykiv-osvitnogo-proczesu.pdf)

1. **Політика щодо відвідування** - відвідування занять є обов'язковим. При наявності поважних причин (хвороба, участь в інших видах навчальної, наукової чи організаційної роботи, офіційна робота за фахом) студенти можуть узгодити з викладачем індивідуальний графік навчання та здачі всіх видів навчальної роботи. Студенти можуть перескладати або відпрацьовувати пропущені заняття на консультаціях викладача чи у спеціально відведений викладачем для цього час.

Здобувач вищої освіти, який має більше 30% пропусків навчальних занять (без поважних причин) від загального обсягу аудиторних годин згідно з індивідуальним начальним планом не допускається до складання екзамену під час семестрового контролю, але має право ліквідувати академічну заборгованість у порядку, передбаченому [Положенням про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка»](https://stu.cn.ua/wp-content/stu-media/normobaza/normdoc/norm-osvitproces/polozhennya-pro-potochne-ta-pidsumkove-oczinyuvannya-znan-zdobuvachiv-vo.pdf)

1. **Політика щодо правил поведінки на заняттях** - активна участь у навчальному процесі, виконання необхідного мінімуму навчальної роботи, коректна поведінка щодо інших учасників навчального процесу, взаємоповага, використання мобільних пристроїв тільки для навчання.
2. **Політика заохочень та стягнень.** Результати навчальної, наукової та організаційної діяльності студентів за напрямами курсу їм можуть нараховуватися додаткові бали - до 10 балів, в залежності від вагомості досягнень студента. Види позанавчальної діяльності, за які студенти заохочуються додатковою кількістю балів: участь у міжнародних проектах, наукові дослідження, тези, статті на науково-практичних конференціях, винаходи, патенти, авторські свідоцтва за напрямами курсу.
3. **Політика оскарження результатів контрольних заходів.** Для вирішення спірних питань, пов’язаних із організацією та проведенням семестрового контролю, оцінювання практик, атестації та визнанні результатів навчання в неформальній та/або інформальній освіті на факультеті створюється апеляційна комісія розпорядженням декана, до складу якої включаються, завідувачі кафедр, науково-педагогічні працівники та представники органів студентського самоврядування. Головою апеляційної комісії призначається декан. Порядок подання та розгляду апеляцій визначається відповідно до р.7 [Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка»](https://stu.cn.ua/wp-content/stu-media/normobaza/normdoc/norm-osvitproces/polozhennya-pro-potochne-ta-pidsumkove-oczinyuvannya-znan-zdobuvachiv-vo.pdf).

**8.** **Рекомендована література та інформаційні джерела:**

1. C Programming Language. Prentice Hall, 1988. 274 p.
2. Python Basics: A Practical Introduction to Python 3 / D. Bader et al. Real Python (realpython.com), 2021. 635 p.
3. Ramalho L. Fluent Python: Clear, Concise, and Effective Programming / ed. by M. Blanchette, R. Roumeliotis. O’Reilly Media, 2015. 792 p.
4. Lutz M. Learning Python. 5th ed. 2013. 1540 p.
5. Barry P. Head First Python: A Brain-Friendly Guide. O'Reilly Media, 2016. 624 p.
6. Modern Operating Systems: Forth Edition. Pearson, 2015.
7. Matthes E. Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming. No Starch Press, 2019. 544 p.
8. Automate the Boring Stuff with Python: Practical Programming for Total Beginners. San Francisco, USA : No Starch Press, 2015. 504 p.
9. Think Python: How to Think Like a Computer Scientist / ed. by M. Blanchette. 2nd ed. Sebastopol, California, United States of America : O’Reilly Media, 2015. 222 p.
10. Slatkin B. Effective Python: 90 Specific Ways to Write Better Python. Addison-Wesley Professional, 2019. 480 p.
11. Gorelick M., Ozsvald I. High Performance Python: Practical Performant Programming for Humans. O'Reilly Media, Incorporated, 2020. 450 p.
12. Система дистанційного навчання НУ «Чернігівська політехніка». Курс: Архітектура комп’ютерних систем. – [Електронний ресурс]. – Режим доступу: <https://eln.stu.cn.ua/course/view.php?id=7199>