

Чернівецький національний університет імені Юрія Федьковича

(повне найменування закладу вищої освіти)

Навчально-науковий інститут фізико-технічних і комп'ютерних наук

(назва інституту/факультету)

Кафедра комп'ютерних систем та мереж

(назва кафедри)

СИЛАБУС

навчальної дисципліни

OK13. Програмування. Ч1. Основи алгоритмізації

(вказати назву навчальної дисципліни (іноземною, якщо дисципліна викладається іноземною мовою))

та програмування на Python і JavaScript

обов'язкова

(обов'язкова чи вибіркова)

Освітньо-професійна програма – “Комп'ютерна інженерія”,

“Програмування мобільних і вбудованих комп'ютерних систем та засобів Інтернету речей”

Спеціальність 123 – Комп'ютерна інженерія

(шифр і назва спеціальності)

Галузь знань 12 – Інформаційні технології

(шифр і назва галузі знань)

Рівень вищої освіти – перший (бакалаврський)

(вказати: перший (бакалаврський)/другий (магістерський)/третій (освітньо-науковий))

Навчально-науковий інститут фізико-технічних і комп'ютерних наук

(назва факультету / інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання – українська

(мова, на якій читається дисципліна)

Кількість кредитів: 6

Форми навчальної діяльності: лекції, лабораторні та практичні роботи, самостійна робота

Форма підсумкового контролю: іспит

Розробники: Гімчинська Сніжана Юріївна, асистент кафедри КСМ

(вказати авторів (викладач (ів)), їхні посади, наукові ступені, вчені звання)

Профайл викладача (-ів) <https://csn.chnu.edu.ua/>,
<https://csn.chnu.edu.ua/employees/gimchynska-snizhana-yuriyivna/>

Контактний тел. +(38) 0372 50 94 32 (кафедра КСМ) – Гімчинська С.Ю.

E-mail: s.gimchynska@chnu.edu.ua

Сторінка курсу в Moodle <https://moodle.chnu.edu.ua/course/view.php?id=2629>

Консультації очні або on-line: згідно з розкладом (1 раз у 2 тижні)

1. Анотація дисципліни

Дисципліна «Програмування. Ч1. Основи алгоритмізації та програмування на Python і JavaScript» призначений для формування компетентностей випускників спеціальності 123 – Комп'ютерна інженерія щодо використання сучасних методів алгоритмізації та програмування для розв'язання задач певного класу. Програма навчальної дисципліни «Програмування. Ч1. Основи алгоритмізації та програмування на Python і JavaScript» передбачає надання здобувачам вищої освіти теоретичних і практичних знань щодо основних понять, методів алгоритмізації та технологій програмування, дослідження властивостей і особливостей алгоритмів, а також процесів створення працездатних програм мовами програмування Python та JavaScript. Отримані знання мають стати важливою конкурентною перевагою у подальшій професійній діяльності.

2. Мета навчальної дисципліни: забезпечення необхідного рівня теоретичної і практичної підготовки студентів для грамотного використання ними знань основ алгоритмізації та програмування при освоєнні суміжних дисциплін та у майбутній професійній діяльності, формування у студентів алгоритмічного мислення та набуття навиків розробки програм мовами програмування Python та JavaScript для вирішення прикладних задач із різних предметних галузей.

Завдання – ознайомлення з основними поняттями алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових структур даних (організація даних); оволодіння процесом проектування і побудови алгоритмів розв'язування прикладних задач мовами програмування Python та JavaScript; вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування; вивчення типових підходів до розробки і аналізу найбільш розповсюджених алгоритмів сортування; формування у студентів спроможності подальшого самостійного засвоєння знань та вмінь.

3. Пререквізити. Вивченню дисципліни передують отримання компетентностей з дисциплін шкільної програми «Інформатика» та «Математика». Вивчення основ алгоритмізації – необхідна умова поєднання теорії і практик програмування, частина математичної культури та загальної культури мислення. Дана дисципліна є необхідною ланкою ланцюга, що забезпечує безперервність математичної та спеціалізованої програмно-алгоритмічної підготовки студентів.

4. Результати навчання

У результаті вивчення навчальної дисципліни студент повинен

4.1. Знати: встановлення та налаштування інтегрованого середовища програмування, написання, налагодження і оптимізації комп'ютерних програм з використанням особливостей і переваг мов програмування Javascript та Python; етапи розробки програм; принципи складання алгоритмів та побудови блок-схем до алгоритмів; форми запису алгоритмів; базові алгоритмічні структури; основні структурні конструкції мов програмування Javascript та Python; правила написання функцій та процедур, модулів; організацію роботи з файлами; визначення принципів процедурного програмування; вивчення типових підходів до розробки і аналізу найбільш розповсюджених алгоритмів сортування.

4.2. Вміти: застосовувати різні описи алгоритмів; використовувати прості й складні умови під час побудови програм; описувати алгоритми розв'язання різних типів задач мовами програмування Javascript та Python; складати й реалізовувати найпростіші лінійні, розгалужені, циклічні алгоритми; використовувати середовища програмування; виконувати аналіз коректності програм, їхнє налагодження та тестування; працювати з технічною літературою, довідниками, стандартами, технічною документацією.

4.3. Набути компетентностей:

ЗК – загальних:

ЗК1. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК2. Здатність вчитися і оволодівати сучасними знаннями.

ЗК3. Здатність застосовувати знання у практичних ситуаціях.

ЗК6. Навички міжособистісної взаємодії.

ЗК7. Вміння виявляти, ставити та вирішувати проблеми.

ЗК8. Здатність працювати в команді.

ЗК10. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.

ФК – фахових (спеціальних):

ФК1. Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК4. Здатність забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки.

ФК5. Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.

ФК6. Здатність проектувати, впроваджувати та обслуговувати комп'ютерні системи та мережі різного виду та призначення.

ФК7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

ФК14. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.

ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

ПРН – програмних результатів навчання:

ПРН1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

ПРН2. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

5. Опис навчальної дисципліни

5.1. Загальна інформація

Назва навчальної дисципліни <i>OK13 Програмування. Ч1. Основи алгоритмізації та програмування на Python і JavaScript</i>												
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових модулів	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	4	1	6	180	2	30	15	-	30	105	-	іспит
Заочна	4	1	6	180	2	8	4	-	8	160	-	іспит

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить (%):

для денної форми навчання – 71,4% ($100\% \times (30 + 15 + 30) / 105$);

для заочної форми навчання – 12,5% ($100\% \times (8 + 4 + 8) / 160$).

5.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	Денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. Основні принципи алгоритмізації та програмування												
Тема 1. Етапи розробки прикладних програм	7	2	-	-	-	5	9	1	-	-	-	8
Тема 2. Алгоритми. Основні поняття алгоритмізації	9	2	-	2	-	5	11	1	-	-	-	10
Тема 3. Базові структури алгоритмів	34	2	4	4	-	20	14	2	-	2	-	10
Тема 4. Сума та добуток числової послідовності. Рекурентні співвідношення	18	2	2	8	-	10	26		2	2	-	22
Разом за ЗМ 1	68	8	6	14	-	40	60	4	2	4	-	50
Змістовий модуль 2. Типові прийоми програмування Python та JavaScript												
Тема 5. Технологія побудови алгоритмів Структурне програмування. Функції Python та JavaScript	22	6	2	4	-	10	46	2	2	2	-	40
Тема 6. Масив як структура даних	18	2	2	4	-	10	10	-	-	-	-	10
Тема 7. Типові алгоритми обробки масивів (сортування, пошук)	16	4	2	-	-	10	10	-	-	-	-	10
Тема 8. Організація обробки файлів	12	2	-	-	-	10	10	-	-	-	-	10
Тема 9. Модульне програмування	19	4	1	4	-	10	34	2	-	2	-	30
Тема 10. Множини	15	2	1	2	-	10	5	-	-	-	-	5
Тема 11. Словники / об'єкти	10	2	1	2	-	5	5	-	-	-	-	5
Разом за ЗМ 2	112	22	9	16	-	65	120	4	2	4	-	110
Усього годин	180	30	15	30	-	105	180	8	4	8	-	160

5.3. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Лінійні алгоритми	2
2	Розгалужені алгоритми	2
3	Циклічні алгоритми	2
4	Підпрограми користувача	2
5	Впорядкування та обробка масивів	3
6	Множини	2
7	Словники / об'єкти	2
	Разом	15

5.4. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Програмування лінійних алгоритмів	2
2	Програмування розгалужених алгоритмів	4
3	Програмування арифметичних циклів	4
4	Програмування ітераційних циклів	2
5	Підпрограми користувача	4
6	Масиви	6
7	Модульне програмування	4
8	Програмування даних типу множина	2
9	Структури даних	2
	Разом	30

5.5. Тематика індивідуальних завдань

В даному курсі виконання індивідуальних завдань не передбачено.*

* ІНДЗ – може бути рекомендовано в окремих випадках для студентів, які успішно освоїли основний навчальний матеріал, з метою поглибленого вивчення чи удосконалення матеріалів певного змістового модуля, або в цілому для навчальної дисципліни за рішенням кафедри чи викладача.

5.6. Самостійна робота

№	Назва теми	К-ть годин
1	Складання і реалізація програм опрацювання табличних величин	10
2	Складання і реалізація рекурсивних підпрограм	10
3	Впорядкування та обробка одновимірних та багатовимірних масивів	12
4	Обробка символічної інформації	10
5	Формування та обробка даних файлового типу. Текстові файли	15
6	Модулі користувача	12
7	Формування та обробка даних типу: словники та об'єкти	10
8	Об'єктно-орієнтоване програмування. Класи. Об'єкти	18
9.	Швидкі (удосконалені) методи сортування: швидке (метод Хоара), метод Шелла, пірамідальне сортування, сортування методом злиття	13
10.	Графіка в Python і задачі моделювання	10
	Разом	120

Примітка. Методичні рекомендації та завдання до лабораторних робіт доступні на інтернет-ресурсах:

навчальний посібник –

https://drive.google.com/drive/folders/1Df8sJTcc4dYEESwTDG_by_RI7YGTALKW

курс «Programming Essentials in Python» мережної академії Cisco – www.netacad.com; <https://lms.netacad.com/course/view.php?id=1511454>

курс «Programming Essentials in JS» мережної академії Cisco – www.netacad.com; <https://lms.netacad.com/course/view.php?id=1102900>

Програмне забезпечення для виконання лабораторних робіт: мови програмування Python та JavaScript.

6. Форми і методи навчання

Форми навчання – це проблемні й оглядові лекції, лабораторні заняття, заняття із застосуванням комп'ютерної та телекомунікаційної техніки, курс мережної академії Cisco «Programming Essentials in Python»; інтерактивні заняття з навчанням одних студентів іншими, інтегровані заняття, відеолекції, відеозаняття і відеоконференції засобами Google Meet, Zoom, Cisco Webex, заняття з використанням системи електронного навчання Moodle.

Методи: проблемний виклад матеріалу, частково-пошукові та дослідницькі лабораторні практикуми, презентації, консультації і дискусії, робота в інтернет-класі: електронні лекції, лабораторні роботи, дистанційні консультації тощо, спрямовані на активізацію і стимулювання навчально-пізнавальної діяльності студентів.

Підходи до навчання: використовуються студентоцентрований, проблемно-орієнтований, діяльнісний, комунікативний, професійно-орієнтований, міждисциплінарний підходи.

Реалізація навчального процесу здійснюється під час лекційних, лабораторних занять, самостійної позааудиторної роботи з використанням сучасних інформаційних технологій навчання, консультацій з викладачами.

Для **формувань умінь та навичок** застосовуються такі **методи навчання**:

- вербальні/словесні (*лекція, пояснення, розповідь, бесіда, інструктаж*);
- наочні (*спостереження, ілюстрація, демонстрація*);
- практичні (*проведення практики*);
- пояснювально-ілюстративний або інформаційно-рецептивний, який передбачає пред'явлення готової інформації викладачем та її засвоєння студентами;
- репродуктивний (*виконання лабораторних завдань за зразком*).

6.1. Словесні методи навчання. Навчальна лекція

За допомогою даного методу забезпечується усне викладення матеріалу великими ємністю й складністю логічних побудов, доказів і узагальнень. В ході лекції використовуються прийоми усного викладення інформації, підтримання уваги протягом тривалого часу, активізації мислення студентів, прийоми забезпечення логічного запам'ятовування, переконання, аргументації, доказів, класифікації, систематизації і узагальнення. В залежності від специфіки лекційного матеріалу іноді використовується лекція-діалог.

6.2. Індуктивний метод навчання

Даний метод навчання використовується в рамках лекційних занять, коли матеріал носить, здебільшого, фактичний характер. В рамках лабораторних занять метод застосовується при виконанні технічних задач, коли студенти використовують раніше здобуті теоретичні знання при роботі з конкретними пристроями (комп'ютерами) та програмними продуктами.

6.3. Репродуктивний метод навчання

Даний метод навчання використовується в рамках лекційних і лабораторних занять, а також під час самостійної роботи студентів. Метод передбачає роботу студентів за визначеним алгоритмом. Згідно з методом для виконання завдань студентам надаються методичні вказівки, правила і навчальні приклади.

6.4. Проблемно-пошукові методи навчання

Проблемно-пошукові методи застосовуються в ході проблемного навчання, а саме в процесі виконання лабораторних робіт та індивідуальних науково-дослідних завдань, де під проблемною ситуацією треба вважати невідповідність між тим, що вивчається і вже вивченим. При використанні проблемно-пошукових методів навчання викладач використовує такі прийоми: створює проблемну ситуацію (ставить питання, пропонує задачу, експериментальне завдання), організує колективне обговорення можливих підходів до рішення проблемної ситуації, стимулює висунування гіпотез, тощо. Студенти роблять припущення про шляхи вирішення проблемної ситуації, узагальнюють раніше набуті знання, виявляють причини явищ, пояснюють їхнє походження, вибирають найбільш раціональний варіант вирішення проблемної ситуації. Викладач обов'язково керує цим процесом на всіх етапах, а також за допомогою запитань-підказок. Також даний метод використовується при опрацюванні матеріалів в системі дистанційної освіти «Moodle».

6.5. Наочний метод навчання

Наочний метод достатньо важливий для студентів, оскільки забезпечує візуальне подання навчального матеріалу, зокрема, з використанням інформаційно-комунікаційних технологій. При викладанні дисципліни наочний метод навчання поєднується зі словесними методами для представлення інформації у вигляді таблиць, рисунків, схем та діаграм.

7. Система контролю та оцінювання

Засобами оцінювання та демонстрування результатів навчання є

- контрольні роботи;
- стандартизовані тести;
- презентації результатів виконаних завдань.

Формами поточного контролю рівня знань є усна та письмова відповідь студента при захисті виконаних лабораторних робіт, кількість отриманих балів при виконанні тестового завдання, а також письмова відповідь при написанні модульних контрольних робіт. Формами підсумкового контролю рівня знань є усна та письмова відповідь студента при здачі іспиту.

7.1. Критерії оцінювання результатів навчання з навчальної дисципліни

Критерієм успішного проходження здобувачем освіти підсумкового оцінювання є досягнення ним мінімальних порогових рівнів оцінок за кожним запланованим результатом навчання навчальної дисципліни.

У залежності від характеру відповіді студента кількість балів за кожний вид діяльності може бути визначена за наступними критеріями:

К-ть балів	Критерії оцінки
Мах	Студент дає вичерпну відповідь на поставлене запитання
0,8 · Мах	Студент при відповіді на поставлене запитання припустився незначних неточностей, які не впливають на суть відповіді
0,6 · Мах	Студент при відповіді на поставлене запитання припустився помилок, які виправляє за допомогою викладача; в середньому може дати правильні відповіді на 50% питань теми
0,4 · Мах	Студент при відповіді на поставлене запитання припустився суттєвих помилок, які все ж таки виправляє за допомогою викладача; дає правильні відповіді на 30% питань теми
0,2 · Мах	Студент за допомогою викладача фрагментарно відповідає на запитання, проте не в повній мірі володіє мінімальним рівнем знань з даного питання
0	Характер відповідей дає підставу стверджувати, що студент неправильно зрозумів суть питання чи не знав правильної відповіді, а тому відповідав, припускаючись грубих помилок.

Примітка: за Мах прийнято максимальну оцінку для даного виду діяльності; заокруглення проводиться до одиниць балу.

Шкала та критерії оцінювання: національна та ЄКТС (Європейська кредитна трансферно-накопичувальна система, ECTS)

Оцінка за шкалою ЄКТС	Пояснення	Оцінка за 100-бальною шкалою	Оцінка за національною шкалою
A	відмінно	90 – 100	відмінно
B	дуже добре	80-89	добре
C	добре	70-79	
D	задовільно	60-69	задовільно
E	достатньо	50-59	
FX	(незадовільно) з можливістю повторного складання	35-49	незадовільно
F	(незадовільно) з обов'язковим повторним курсом	1-34	

Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота													Підсумковий контроль (іспит)	Сумарна кількість балів
Змістовий модуль 1					Змістовий модуль 2									
T1	T2	T3	T4	M1	T5	T6	T7	T8	T9	T10	T11	M2	40	100
2	5	5	8	8	5	5	2	-	4	3	5	8		

T1, T2 ... T11 – теми змістових модулів; M1, M2 – модульні контрольні роботи

7.2. Перелік тем та розподіл максимально можливої кількості балів, які отримують студенти за виконання всіх видів навчальної діяльності

Змістовий модуль 1. Основні принципи алгоритмізації та програмування

- T1. Етапи розробки прикладних програм (тест №1 – 2 бали)
- T2. Алгоритми. Основні поняття алгоритмізації (тест №2 – 2 бали; лаб. робота №1 – 3 бали)
- T3. Базові структури алгоритмів (тест №3 – 2 бали; лаб. робота №2 – 3 бали)
- T4. Сума та добуток числової послідовності. Рекурентні співвідношення (тест №4 – 2 бали; лаб. роботи №3, №4 – по 3 бали)
- M1. Модульна контрольна робота №1 – 8 балів

Змістовий модуль 2. Типові прийоми програмування Python та JavaScript

- T5. Технологія побудови алгоритмів
 - T5.1. Структурне програмування (тест №5 – 2 бали)
 - T5.2. Функції Python та JavaScript (лаб. робота №5 – 3 бали)
- T6. Масив як структура даних (тест №6 – 2 бали; лаб. робота №6 – 3 балів)
- T7. Типові алгоритми обробки масивів (сортування, пошук) (тест №7 – 2 бали)
- T8. Організація обробки файлів

- T9. Модульне програмування (лаб. робота №7 – 4 балів)
 T10. Множини (лаб. робота №8 – 3 балів)
 T11. Словники / об'єкти (тест №8 – 2 бали; лаб. робота №9 – 3 балів)
 M2. Модульна контрольна робота №2 – 8 балів

7.3. Умови зарахування результатів неформальної освіти

Студент, згідно Положення ЧНУ «Про неформальну освіту» може отримати додаткові бали, або бути звільненим від окремих видів роботи з окремих тем, якщо у нього наявні сертифікати про неформальну освіту з проблем, які вивчаються на дисципліні «Програмування. Ч1. Основи алгоритмізації і програмування на Python і JavaScript».

Також, як виконані види роботи з відповідних тем зараховуються студенту бали за наукові публікації у матеріалах науково-практичних конференцій та фахових чи апробаційних виданнях.

7.4. Політика курсу

Самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей).

Академічна доброчесність: посилання на джерела інформації у разі використання ідей, розробок, тверджень, відомостей; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерела інформації.

Відвідування: Відвідування занять є обов'язковим. Засвоєння пропущеної теми лекції з поважної причини перевіряється під час складання підсумкового контролю. Пропуск лекції з неповажної причини відпрацьовується студентом (співбесіда, реферат тощо). Пропущені практичні та лабораторні заняття, незалежно від причини пропуску, студент відпрацьовує згідно з графіком консультацій.

8. Рекомендована література

Фахова (основна)

1. Гімчинська С.Ю. Основи алгоритмізації та програмування (Python та JavaScript) : Навчальний посібник. Видання 2-е, доп. та переробл. / Чернівці : Чернівецький національний університет ім. Ю. Федьковича, 2022. 200 с.
2. Обчислювальна техніка та програмування : контрольні та тестові завдання / укл. Гімчинська С.Ю. – Чернівці : Чернівецький національний університет, 2010. – 60 с.
3. Основи алгоритмізації та програмування (Python та JavaScript) : Методичні вказівки до практичних занять / Чернівці : Чернівецький національний університет імені Юрія Федьковича, 2022. 90 с. (електронне видання).
4. Азарян А.А., Карабут Н.О., Козикова Т.П., Рибальченко О.Г., Трачук А.А., Шаповалова Н.Н. Основи алгоритмізації та програмування: Навчальний посібник. – Кривий Ріг: Вид-во ОктаПринт, 2014. – 308 с.
5. Пол Беррі. Python. Легкий для сприйняття довідник. Вид-во Фабула, 2021. - 624 с.
6. Кренивч А.П. Python у прикладах і задачах. Частина 1. Структурне програмування Навчальний посібник із дисципліни "Інформатика та програмування" – К.: ВПЦ "Київський Університет", 2017. – 206 с.

7. Саммерфилд М. Python на практике. / Марк Саммерфилд – М.: ДМК Пресс, 2014. – 338 с.
8. Маккинли У. Python и анализ данных. / У. Маккинли – М.: ДМК Пресс, 2015. – 482 с.
9. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині"/ А.В. Яковенко; КПІ ім. Ігоря Сікорського. –Київ: КПІм. Ігоря Сікорського, 2018. – 195 с.
10. Програмування числових методів мовою Python : підруч. / А.В. Анісімов, А.Ю. Дорошенко, С.Д. Погорілий, Я.Ю. Дорогий ; за ред. А.В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.
11. Илья Кантор. Современный учебник JavaScript. / learn.javascript.ru, 2015. – 400 с.
12. Ерік Маттес. Пришвидшений курс Python. Практичний, проектно-орієнтований вступ до програмування. – Видавництво Старого Лева, 2021. – 600 с.
13. Олексій Васильєв. Програмування мовою Python. – Видавництво Навчальна книга – Богдан, 2019. – 504 с.

Допоміжна

1. Кириленко Н.М. Основи алгоритмізації та програмування : Лабораторний практикум: навч. посіб. / Н. М. Кириленко. – Вінниця : – ГЛОБУС-ПРЕС, 2014. – 200 с.
2. Співаковський О.В., Осипова Н.В., Львов М.С., Бакуменко К.В. Основи алгоритмізації та програмування. Обчислювальний експеримент. Розв'язання проблем ефективності в алгоритмах пошуку та сортування: Навчальний посібник. – Херсон: Айлант. – 2011. – 100 с.
3. JavaScript: The Definitive Guide. 7th Ed. David Flanagan (english). Видав. – O'Reilly Media, 2020. – 600 с.

9. Інформаційні ресурси

1. Система підтримки дистанційного навчання "Moodle".
2. www.netacad.com
3. <https://www.python.org/> (Офіційний сайт мови програмування Пітон).
4. <https://metanit.com/web/javascript/4.10.php>