



ІНФОРМАТИКА

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>13 – Механічна інженерія</i>
Спеціальність	<i>131 Прикладна механіка</i>
Освітня програма	<i>Конструювання та дизайн машин</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)/дистанційна/змішана</i>
Рік підготовки, семестр	<i>2 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити ЕКТС, 120 год., Лекції – 18 год., лабораторні – 54 год., СРС 48 год.</i>
Семестровий контроль/ контрольні заходи	<i>Залік, МКР</i>
Розклад занять	<i>За розкладом на сайті університету: http://roz.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор доцент кафедри КМ, к.т.н., Івановський Олексій Анатолійович Лабораторні: , к.т.н., Івановський Олексій Анатолійович ivanovskiy.oleksii@i11.kpi.ua, к.т.н. Парненко В.С.</i>
Розміщення курсу	<i>Ресурс «Електронний кампус»</i>

Програма навчальної дисципліни

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

Дисципліна «Інформатика» є нормативною для підготовки бакалаврів за освітньою програмою «Конструювання та дизайн машин».

Дана дисципліна передбачає вивчення об'єктно-орієнтованої мови програмування Python, бібліотеки стандартних модулів і принципів розробки програмних систем.

Метою дисципліни "Інформатика" є формування у студентів навичок, відповідних видів професійної діяльності, на які орієнтована програма, необхідних для вирішення наступних професійних завдань:

- побудова математичних моделей і дослідження їх аналітичними методами,
- розробка алгоритмів, методів, програмного забезпечення, інструментальних засобів по тематиці проведених науково-дослідних проектів
- розробка і застосування сучасних високопродуктивних обчислювальних технологій, застосування сучасних суперкомп'ютерів у випробуваннях
- розробка архітектури, алгоритмічних і програмних рішень системного і прикладного програмного забезпечення;
- розвиток і використання математичних та інформаційних інструментальних засобів, автоматизованих систем в науковій і практичній діяльності;
- застосовувати професійні знання й уміння на практиці;
- адаптуватися до різних професійних ситуацій, проявляти творчий підхід, ініціативу;
- аналізувати проблеми, ставити постановку цілей і завдань, виконувати вибір та вирішення проблем.

Вивчення навчальної дисципліни «Інформатика» дозволить використовувати знання основ програмування при розробці інформаційних технологій, що зараз охоплюють майже всі сфери життя і діяльності та відіграють важливу.

У результаті вивчення дисципліни студент набуде наступних компетентностей:

Загальні компетентності

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК9. Навички використання інформаційних і комунікаційних технологій.

Завершитись навчання повинно наступними програмними результатами:

РН1. Вибирати та застосовувати для розв'язання задач прикладної механіки придатні математичні методи.

РН8. Знати і розуміти основи інформаційних технологій, програмування, практично використовувати прикладне програмне забезпечення для виконання інженерних розрахунків, обробки інформації та результатів експериментальних досліджень.

РН11. Розуміти принципи роботи систем автоматизованого керування технологічним обладнанням, зокрема мікропроцесорних, вибирати та використовувати оптимальні засоби автоматизації.

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

Навчальна дисципліна «Інформатика» має тісний зв'язок з навчальними дисциплінами «Вища математика. Частина 1. Диференціальне та інтегральне числення функції однієї змінної», «Вища математика. Частина 2. Диференціальне та інтегральне числення функції багатьох змінних. Диференціальні рівняння».

Дана дисципліна забезпечує такі дисципліни, як «Механіка матеріалів і конструкцій. Курсова робота», «Теорія механізмів і машин», «Теорія механізмів і машин. Курсова робота», «Деталі машин і основи конструювання», «Деталі машин і основи конструювання. Курсовий проект» та інші дисципліни професійної підготовки відповідно до освітніх програм.

3. Зміст навчальної дисципліни

Розділ 1 Вступ.

Тема 1.1. Типізація в мовах програмування. Области застосування Python, Структура програми. Введення в інтерпретатор. Користувальницький інтерфейс IDLE

Розділ 2. Основні типи даних

Тема 2.1. Числа. Арифметичні операції з числами. Модуль math. Списки. Кортежі. Словники

Тема 2.2. Умови. Логічні операції. Типи даних

Розділ 3. Цикли

Тема 3.1. Цикл. Налаштування програми. Завдання на цикли. Прийоми програмування циклів.

Розділ 4. Функції в програмуванні

Тема 4.1 Параметри і аргументи функцій. Локальні і глобальні змінні. Програмування зверху вниз.

Тема 4.2. Рекурсивний виклик функції

Розділ 5. Модулі

Тема 5.1. Створення і використання модулів.

Розділ 6. Класи і об'єктно-орієнтоване програмування

Тема 6.1. Методи, успадкування, інструкції.

Тема 6.2. Об'єктно-орієнтоване програмування, класи, об'єкт, екземпляри класу

4. Навчальні матеріали та ресурси

Основна література:

1. Олексій Васильєв, Програмування мовою Python / Навчальна книга – Богдан Київ - 2019 480 с. https://opac.kpi.ua/F/?func=direct&doc_number=000637846&local_base=KPI01
2. Висоцька, Вікторія Анатоліївна. PYTHON : Алгоритмізація та програмування : навчальний посібник / В.А. Висоцька, О.В. Оборська ; Міністерство освіти і науки України, Національний університет "Львівська політехніка". – Львів : Видавництво "Новий Світ-2000", 2021. – 514с. https://opac.kpi.ua/F/?func=direct&doc_number=000637149&local_base=KPI01
3. Маттес, Ерік. Пришвидшений курс Python : практичний, проектно-орієнтований вступ до програмування / Ерік Маттес ; з англійської переклала Ольга Белова. – Львів : Видавництво Старого Лева, 2021. – 556 с. https://opac.kpi.ua/F/?func=direct&doc_number=000633837&local_base=KPI01
4. Мельник, Ігор Віталійович. Основи програмування на мові Python : комплексний навчальний посібник : в 2 томах / І.В. Мельник. - Київ : Кафедра, 2020. – 2 т. – Том 1, Базові принципи побудови мови програмування Python та її головні синтаксичні конструкції. – 2020. – 372 с. Том 2, Розвинені засоби мови програмування Python. – 2020. – 491 с. https://opac.kpi.ua/F/?func=direct&doc_number=000633555&local_base=KPI01

Додаткова література:

1. Яковенко, А. В. Основи програмування. Python. Частина 1 [Електронний ресурс] : підручник для студентів які навчаються за спеціальністю 122 «Комп'ютерні науки» спеціалізацією «Інформаційні технології в біології та медицині» / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1файл: 1,71 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с. <https://ela.kpi.ua/handle/123456789/25111>
2. Основи програмування [Електронний ресурс] : методичні вказівки до виконання комп'ютерних практикумів на PYTHON з навчальної дисципліни «Основи програмування» для студентів спеціальності 122 «Комп'ютерні науки» зі спеціалізації «Інформаційні технології в біології та медицині» / Національний технічний університет України «Київський політехнічний інститут» ; укладач Л. М. Добровська ; редактор А. В. Яковенко. – Київ : КПІ ім. Ігоря Сікорського, 2017. – 254с. <http://ela.kpi.ua/handle/123456789/19094>

Інформаційні ресурси:

1. <https://docs.python.org/uk/3/tutorial/index.html> (Електронний підручник з Python)
2. https://youtube.com/playlist?list=PLuVU0VikXlkmR5OO_nkbx8FYm8Iznzy6M (Відео курс Івановського Олексія Python(Beginning))
3. <https://wombat.org.ua/AByteOfPython/AByteofPython-2.02.pdf> (Електронний підручник з Python)
4. <https://github.com/alex1980-m/Tutorial-for-Python> (Онлайн курс та завдання Івановського Олексія)

Наведена література знаходиться в бібліотеці КПІ ім. Ігоря Сікорського (<https://ela.kpi.ua/>) та в мережі Internet.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

На лекціях подається теоретичний матеріал та наводяться приклади розв'язування основних тематичних задач. Під час лекційних занять розглядаються наступні питання:

№ з/п	Назва теми лекції та перелік основних питань
1	<p><i>Тема 1.1. Типізація в мовах програмування. Области застосування Python, Структура програми. Введення в інтерпретатор. Користувальницький інтерфейс IDLE</i></p> <p>Лекція 1. Структура програмами. Робота з документацією. Інтегроване середовище розробки. Перша програма. Додаткові можливості IDLE, інші інтегровані середовища розробки, парадигми програмування, запуск модулів за допомогою функцією exes.</p>
2	<p><i>Тема 2.1. Числа. Арифметичні операції з числами. Модуль math. Списки. Кортежі. Словники</i></p> <p>Лекція 2. Базові числові типи. Числові розширення. Двійкові оператори. Оператори для роботи з послідовностями. Пріоритет виконання операторів. Списки в дії. Переглянутий перелік категорій файлів. Гнучкість об'єктів. Ієрархії типів даних. Пастки вбудованих типів. Посилання та копії. Словники. Модуль Turtle. Бот і праця з ним.</p>
3	<p><i>Тема 2.2. Умови. Логічні операції. Типи даних</i></p> <p>Лекція 3. Умовні інструкції if , перевірка істинності, тримісне вираз if/else , синтаксичні правила мови Python, масиви.</p>
4	<p><i>Тема 3.1 . Цикл. Налаштування програми. Завдання на цикли. Прийоми програмування циклів</i></p> <p>Лекція 4. Цикли while, break, continue. Цикли for. Прийоми програмування циклів. Налаштування програми. Функція range. Вкладені цикли. Ітератори, генератори списків, нові об'єкти, файли та перетворення їх в модулі.</p>
5	<p><i>Тема 4.1. Параметри і аргументи функцій. Локальні і глобальні змінні. Програмування зверху вниз</i></p> <p>Лекція 5. Области видимості в мові Python. Інструкція global, nonlocal. Створення функцій. Передача аргументів. Спеціальний режим зіставлення аргументів. Універсальні функції для роботи з безліччю.</p>
6	<p><i>Тема 4.2. Рекурсивний виклик функції</i></p> <p>Лекція 6 Концепції проектування функції. Функція пошуку мінімуму. Універсальні функції для роботи з безліччю. Розробка інтерфейсу за допомогою модуля(Tkinter.</p>
7	<p><i>Тема 5.1. Створення і використання модулів.</i></p> <p>Лекція 7. Транзитивне перезавантаження модуля і метапрограми. Розширення as для інструкції import і from. Розробка інтерфейсу з Kivy.</p>
8	<p><i>Тема 6.1 Методи, успадкування, інструкції.</i></p> <p>Лекція 8. Простори назв: закінчення історії. Множинне спадкування: домішкові класи. Методи - це об'єкти: зв'язані і незв'язані методи. Статичні методи і методи класу. ООП і делегування: об'єкти-обгортки.</p>
9	<p><i>Тема 6.2. Об'єктно-орієнтоване програмування, класи , об'єкт, екземпляри класу</i></p> <p>Лекція 9. ООП і делегування: об'єкти-обгортки. Зміни в класах нового стилю. Конструктори , класи або статичні змінні, створення екземплярів класу.</p>

Лабораторні роботи

На лабораторних роботах студенти опановують методики базового програмування, працювання з стандартними функціями , правилами запису математичних та логічних виразів, вчать розробляти сучасні методи розробки сценаріїв для інженерних програм (САПР), розробляти інтерфейси які використовують в математичних та механічних розрахунках. Дані роботи допоможуть

студентам в навчання та розробки машинного навчання та нейронних мереж які широко використовуються у зв'язці з САПР програмами.

Тематика лабораторних робіт:

№ №	Назва і зміст	З якою темою пов'язана
1	Вступне заняття. Видача варіантів завдань. Рядки. Ознайомлення студентів з операторами присвоєння, стандартними функціями, правилами запису математичних та логічних виразів.	1.1
2	Робота з базовими числовими типами. Ознайомитися з основними типовими прийомами програмування – обчислення значень сум та добутків та навчитися їх застосовувати на практиці.	1.1
3	Розробка списку . Ознайомитися з основними функціями та стандартними процедурами обробки стрічок та навчитися застосовувати їх для написання програм	2.2
4	Праця з графічним модулем Turtle. Навчитися працювати з вбудованим модулем , та почати програмувати графічні об'єкти.	2.2
5	Розробка словника. Обробка тексту та перетворення його на словник.	2.2
6	Створення першого бота на основі даного модуля. Розробка модуля , та підключення його до сучасних пакетів комунікації , перша робота з токенами та запуск готової програми.	2.2
7	Створення модулів. Ознайомлення студентів з використанням модулів, особливостями їх побудови, вивчення методів компіляції.	3.1
8	Створення кортежа та перетворення його в цифровий код. Навчитись працювати з об'єктами динамічної природи: динамічними змінними; ознайомитися з основними процедурами для роботи з динамічними змінними .	2.3
9	Створення циклів. Розглянути циклічні алгоритми і програми та реалізувати запропоновані програми за допомогою циклів з передумовою та післяумовою та оператора for, if, else.	3.1
10	Функції . Ознайомитися з особливостями роботи підпрограм, навчитися створювати підпрограми-процедури та процедури-функції, зробити висновки щодо доцільності використання підпрограм.	2.2
11	Розробка та праця з масивами . Ознайомитися з особливостями організації одновимірних масивів, навчитися виконувати обробку одновимірних масивів, здійснювати сортуванням масивів.	2.2
12	Масиви, створення модуля. Ознайомитися з роботою двовимірних масивів, навчитися здійснювати сортуванням масиву та виконувати найпростіші дії над елементами масиву.	2.2
13	Створення функції и графіків. Написати основні функції за допомогою def та стандартними процедурами обробки та навчитися застосовувати їх для написання програм.	4.1
14	Розробка інтерфейсу з Tkinter. Навчитись розробляти різні елементи керування (такі як мітки, кнопки, поля зі списком, прапорці, панелі меню, радіокнопки та багато іншого) для взаємодії з програмою.	4.2
15	Розробка базових елементів інтерфейсу з Kivy. Навчитись розробляти різні елементи керування (такі як мітки, кнопки, поля зі списком, прапорці, панелі меню) для взаємодії з програмою.	5.1
16	Моделювання MessageBox, Dropdown, та праця з розкритим списком . Це дозволяє відображати список віджетів під відображеним віджетом та навчитись розробляти їх.	5.1

№ №	Назва і зміст	З якою темою пов'язана
17	Моделювання за допомогою різноманітних елементів та віджетів інтерфейса	4.2, 5.1
18	Праця за класами. Створення програми для імпорту вбудованого модуля масиву та відображення простору імен зазначеного модуля. Це дозволяє зрозуміти алгоритм дії.	6.1
19	Написати програму Python для створення класів та відображення простору імен зазначеного класу. Навчитись працювати з об'єктно-орієнтованим програмуванням (ООП), використовувати екземпляри класу, об'єкти та різноманітні методи .	6.2
20	Написати програму Python, яка приймає список цілих чисел і перевіряє довжину та п'ятий елемент. Повертає true, якщо довжина списку дорівнює 8 і п'ятий елемент зустрічається у цьому списку тричі. Ознайомитися з особливостями ООП, навчитися виконувати перевантаження функцій на прикладі математичних дій.	4.1, 4.2, 5.1
21	Складання програми Python GUI для імпорту пакета Tkinter, створення вікна та встановлення його заголовка. Праця з інтерфейсом та підключення його для розробленого модуля.	4.1,4.2, 5.1
22	Розробка програми Python GUI для створення віджетів панелі виконання за допомогою модуля Kivy. Праця з різноманітними інтерфейсами та сторонніми модулями.	4.2, 5.1, 6.1
23	Розробка комплексної програми з інтерфейсом та вбудованою функцією, використовуючи графічний модуль . Використання особливостей об'єктно-орієнтованого програмування в зв'язці з інтерфейсом.	3.1,4.1,4.2,5.1 , 6.2
24	Розробка програми Python для генерації числа з плаваючою точкою від 0 до 1 включно та генерування випадкового числа з плаваючою точкою в межах певного діапазону. Використання модуля random в об'єктно-орієнтованому програмуванні.	2.1,2.2,3.1, 5.1.
25	Створення комплексного завдання яке включає всі елементи програмування вивчені на протязі курсу. Яке повинно включати елементи інтерфейсу, об'єктно-орієнтоване програмування, модулі, токени , екземпляри , класи , функції.	1.1,2.1,2.2,3.1, 4.1,4.2,5.1,6.1, 6.2

6. Самостійна робота студента

Години, відведені на самостійну роботу студента, призначені для опанування навчальної дисципліни, зокрема, підготовка до виконання робіт на лабораторних заняттях; підготовка до лекцій, модульної контрольної роботи та заліку.

Перелік завдань для виконання в межах часу самостійної роботи наведено в п. 9.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- правила відвідування занять (як лекцій, так і практичних/лабораторних) регламентується: «Положення про організацію освітнього процесу в КПІ ім. Ігоря Сікорського» <https://osvita.kpi.ua/node/39>; «Положення про систему внутрішнього забезпечення якості вищої освіти в КПІ ім. Ігоря Сікорського» <https://osvita.kpi.ua/node/121>;

- правила поведінки на заняттях (активність, підготовка коротких доповідей чи текстів, відключення телефонів, використання засобів зв'язку для пошуку інформації на гугл-диску викладача чи в інтернеті тощо) регламентується «Положення про організацію освітнього процесу в КПІ ім. Ігоря Сікорського» <https://osvita.kpi.ua/node/39>, ;
- правила захисту індивідуальних завдань; кожен студент особисто здає індивідуальні роботи;
- в даному кредитному модулі наявні тільки заохочувальні бали, які студент може отримати на добровільній основі виконуючі певний перелік додаткових завдань пов'язаних з тематикою кредитного модуля;
- політика дедлайнів та перескладань, регламентується «Положення про поточний, календарний та семестровий контроль результатів навчання в КПІ ім. Ігоря Сікорського» <https://osvita.kpi.ua/node/32>, «Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського» <https://osvita.kpi.ua/node/37>;
- політика щодо академічної доброчесності регламентується «Положення про систему запобігання академічного плагіату в КПІ ім. Ігоря Сікорського» <https://osvita.kpi.ua/node/47>; положенням «Положення про вирішення конфліктних ситуацій в КПІ ім. Ігоря Сікорського» https://osvita.kpi.ua/2020_7-170;
- З метою вивчення дисципліни під час лекційних та практичних занять необхідно використовувати підручники, посібники, практикувати навички розв'язування реальних задач. Проведення індивідуального консультування відбувається протягом курсу навчання.
- Для успішного засвоєння курсу слід передбачається тісний взаємозв'язок всіх видів занять - лекційних, лабораторних та індивідуальних. Теоретичний матеріал, викладений на лекційних заняттях є основою для вирішення завдань, що виконуються на лабораторних заняттях. Це дозволяє поглибити знання з кожної теми.
- Захист лабораторних робіт відбувається після виконання всіх завдань отриманих на заняттях. Включає в себе як звіт по практичній частині так і відповіді на теоретичні питання.

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

Поточний контроль: лабораторні роботи, модульна контрольна робота.

Календарний контроль: провадиться 2 рази на семестр за встановленим графіком як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік.

Рейтингова оцінка R студента з дисципліни складається з балів, які він отримує за:

- виконання лабораторних робіт r_1 ;
- модульних контрольних робіт r_2 ;
- залік r_3 .

Додатково PCO передбачає можливість нарахування заохочувальних та штрафних балів.

Лабораторні роботи r_1

Необхідною умовою допуску до лабораторної роботи є наявність протоколу. Ваговий бал однієї лабораторної роботи становить 2,4 бали. Максимальна кількість балів за всі лабораторні роботи: $r_1 = 25 \text{ робіт} \times 2,4 \text{ бали} = 60 \text{ балів}$.

Рейтингові бали за одну лабораторну роботу

Оцінка	Бали	Критерій оцінювання
A	2,4	Зауважень до звіту нема, є відповіді на всі запитання
B	2,2	Несуттєві зауваження до звіту, відповіді на більшість запитань
C	1,9	Зауваження до отриманих результатів, відповідь на частину питань
D	1,7	Звіт має помилки, відповіді лише на окремі питання
E	1,4	Робота виконана, отримано вірні результати, але не захищено.

<i>F_x</i>	0	Робота не виконана, звіт відсутній
----------------------	---	------------------------------------

Модульний контроль r_2

Робочим навчальним планом передбачено проведення однієї модульної контрольної роботи (МКР) в обсязі 2 год. МКР відбувається у вигляді чотирьох контрольних робіт по півгодини кожна. Контрольна робота-1 виконується за розділами 1, 2. Контрольна робота-2 виконується за розділами 3 і 4. Контрольна робота-3 виконується за розділом 5. Контрольна робота-4 виконується за розділом 6.

Одна контрольна робота складається з тестів які складаються з 20 питань та завдань. Завдання оновлюються кожного семестру. Ваговий бал однієї контрольної роботи – 10 балів.

Максимальна кількість балів за чотири контрольні роботи складає: $r_2 = 10 \text{ балів} \times 4 \text{ контр.роботи} = 40 \text{ балів}$.

Рейтингові бали за одну КР

Оцінка	Бали	Критерій оцінювання
A	10	Вірна відповідь більш ніж на 90 % питань
B	9	Вірна відповідь на 90 % питань
C	8	Вірна відповідь на 80 % питань
D	7	Вірна відповідь на 70 % питань
E	6	Вірна відповідь на 60 % питань
<i>F_x</i>	0	Вірна відповідь менш ніж на 60 % питань або студент був відсутній

Штрафні та заохочувальні бали

Загальний рейтинг з дисципліни включає штрафні та заохочувальні бали, які додаються до суми вагових балів усіх контрольних заходів.

Нарахування штрафних балів не передбачено.

Заохочувальні бали можуть нараховуватися за виконання творчих робіт: робота у наукових гуртках з підготовкою матеріалів доповідей або статей для публікації, участь у наукових і науково-практичних конференціях і семінарах, олімпіадах з дисципліни, конкурсах робіт, рефератів та оглядів наукових праць, аналіз сучасної нормативно-правової бази з дисципліни у країні та її відповідність вимогам міжнародних стандартів тощо. Кількість нарахованих балів залежить від отриманих результатів.

Загальна сума заохочувальних балів не може перевищувати 10% від рейтингової шкали, тобто $100 \times 0,1 = 10 \text{ балів}$.

Умови рубіжної атестації

Календарний контроль з навчальної дисципліни (освітнього компонента) проводиться, як правило, на 7-8 та 14-15 тижнях кожного семестру. Умовою отримання позитивної оцінки з календарного контролю з навчальної дисципліни є значення поточного рейтингу здобувача не менше, ніж 50 % від максимально можливого на час проведення такого контролю. Результати календарного контролю заносяться у модуль «Календарний контроль» Електронного кампусу.

Критерії залікового оцінювання

Рейтингова система оцінювання складається з балів, отриманих здобувачем за результатами заходів поточного контролю, заохочувальних та штрафних балів. Рейтингова оцінка доводиться до здобувачів на останньому занятті з дисципліни в семестрі.

Необхідною умовою допуску до заліку є виконання та захист всіх лабораторних робіт, написання рубіжних тестів.

Рейтингова оцінка доводиться до здобувачів на передостанньому занятті з дисципліни в семестрі. Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідно до набраного рейтингу оцінку без додаткових випробувань.

Зі здобувачами, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також з тими здобувачами, хто бажає підвищити свою рейтингову оцінку, на останньому занятті з дисципліни в семестрі викладач проводить семестровий контроль у вигляді залікової контрольної роботи.

У цьому випадку бали, отримані за індивідуальну роботу залишаються, а бали отримані за модульні контрольні роботи скасовуються.

Максимальна кількість балів, отриманих за залікову контрольну роботу, складає 40 балів:

$$r_3 = 40 \text{ балів.}$$

Критерій залікового оцінювання визначається як сума якості відповідей на всі завдання білета:

Кількість балів за всі завдання білета

Бали	Критерій оцінювання
40	Відмінна відповідь (не менше 95% інформації), можливі несуттєві зауваження та неточності
36	Дуже добра відповідь (не менше 85% інформації), помилок немає, відповідь на переважну більшість питань, творче мислення
32	Добра відповідь (не менше 75% інформації), помилок немає, відповідь на більшість питань, окремі недоліки
28	Задовільна відповідь (не менше 65% інформації) є зауваження, відповідь на частину питань
24	Достатня відповідь (не менше 60% інформації), суттєві помилки, відповідь на окремі питання.
0,0	Відповідь невірна або менше 60% інформації, або вона відсутня

Розрахунок шкали рейтингу з дисципліни

За результатами заходів поточного контролю з дисципліни, заохочувальних балів

- без залікової контрольної роботи:

$$R = r_1 + r_2 = 60 + 40 = 100 \text{ балів}$$

- із заліковою контрольною роботою:

$$R = r_1 + r_3 = 60 + 40 = 100 \text{ балів}$$

Для отримання відповідної оцінки з дисципліни студент має набрати певну кількість балів, згідно з таблицею перерахунку:

Таблиця перерахунку рейтингових балів в оцінки

Рейтингова оцінка здобувача	Університетська шкала оцінок рівня здобутих компетентностей
95 ... 100	Відмінно
85 ... 94	Дуже добре
75 ... 84	Добре
65 ... 74	Задовільно
60 ... 64	Достатньо
Менше 60 балів	Незадовільно
Не виконані умови допуску до семестрового контролю	Не допущено

9. Додаткова інформація з дисципліни

Для перевірки студентами отриманих знань і навичок їм пропонуються наступні завдання для виконання в межах часу самостійної роботи:

1. Створення першої програми та праця з інтерпретатором.
2. Зробити геометричну фігуру за допомогою модуля turtle
3. Зв'яжіть будь-яку змінну з рядком: Ми обов'язково навчимося програмувати . Вийміть з неї такі зрізи: виведіть символи з індексами, кратними трьом.
4. Створіть два списки в діапазоні (0, 100) з кроком 10. Дайте деяким змінним значення цих списків.
5. Створіть два кортежу: один з чисел в діапазоні (1, кількість студентів в групі) з кроком 1, другий - з прізвищ студентів вашої групи. Нехай вони відповідають один одному;
6. n дітей грають в наступну гру. Перед ними гора з m кубиків, перша дитина виймає з купи 1 кубик, кожний наступний дитина - в два рази більше попереднього і так по колу. Якщо число кубиків, які потрібно вийняти, перевищує 25, з нього віднімається 25 і відлік і йде від зменшеного числа, наприклад, замість 32 кубиків буде вийнято 7, потім 14 і т. д. Тим, хто програв вважається той, хто не зміг витягнути необхідну кількість кубиків (в купі залишилося недостатньо). Визначте хто програв.(цикли)
7. Створіть і виведіть на екран масиви. Утворені матриці збережіть в текстові файли. З нулів одновимірні довжини 10 і 55, матрицю розмірами 3×4 , тривимірний масив форми $2 \times 4 \times 5$;
8. Побудуйте графік загасаючої синусоїди $e^{-x} \sin(2x)$ на відрізку $[0; 10]$, використовуючи крок по абсциссе, рівний 0.1.
9. Перебудуйте графіки так, щоб кожна крива розташовувалася на одному графіку за допомогою команди subplot, легенду приберіть, а її текст перемістіть в назву відповідного графіка. Графіки розташуйте на полотні в один стовпець.
10. Вирішіть систему лінійних рівнянь, матрицю коефіцієнтів і стовпець вільних членів прочитайте з текстових файлів. Запишіть в новий текстовий файл отримані коріння.
11. Згенеруйте випадкові вектори з n дійсних значень з рівномірним і нормальним розподілом, а також з n цілих чисел.
12. Згенеруйте випадковий процес, який представляє собою суму рівномірно розподілених на відрізку $[10; 10]$ випадкових величин і нормально розподілених випадкових величин з параметрами (0; 1), довжиною в 10000 значень. Побудуйте гістограму його розподілу.
13. Напишіть програму Python для обчислення довжини рядка
14. Напишіть програму на Python, щоб отримати рядок із перших 2 і останніх 2 символів заданого рядка. Якщо довжина рядка менше 2, повертається замість порожнього рядка.
15. Напишіть програму на Python, щоб змінити заданий рядок на новий рядок, де перший і останній символи були поміняні місцями
16. Напишіть програму Python для видалення символів, які мають непарні значення індексу заданого рядка
17. Напишіть сценарій Python, який приймає дані від користувача та відображає їх у верхньому та нижньому регістрах
18. Напишіть програму Python, яка приймає послідовність слів, розділених комами, як вхідні дані та друкує унікальні слова в сортованому вигляді (буквено-цифровий)
19. Напишіть функцію Python, щоб вставити рядок у середину рядка
20. Напишіть функцію Python, щоб отримати рядок із 4 копій двох останніх символів зазначеного рядка (довжина має бути принаймні 2)
21. Напишіть функцію Python, щоб отримати рядок із перших трьох символів зазначеного рядка. Якщо довжина рядка менша за 3, поверніть вихідний рядок.
22. Напишіть програму на Python для підрахунку повторів кожного слова в в будь-якому реченні
23. Напишіть програму Python для підсумовування всіх елементів у списку
24. Напишіть програму Python для множення всіх елементів у списку

25. Напишіть програму Python, щоб отримати найбільше число зі списку
26. Напишіть програму Python, щоб отримати найменше число зі списку
27. Напишіть програму на Python для підрахунку кількості рядків, у яких довжина рядка становить 2 або більше, а перший і останній символи однакові з заданого списку рядків.
28. Напишіть програму Python, щоб отримати список, відсортований у порядку зростання за останнім елементом у кожному кортежі з заданого списку непорожніх кортежів
29. Напишіть програму Python для видалення дублікатів зі списку `a = [10,20,30,20,10,50,60,40,80,50,40]`
30. Напишіть програму на Python, щоб перевірити, порожній список чи ні
31. Напишіть програму Python для клонування або копіювання списку
32. Напишіть програму на Python, щоб знайти список слів, які довші за `n` із заданого списку слів.
33. Напишіть функцію Python, яка приймає два списки та повертає `True`, якщо вони мають принаймні один спільний член
34. Напишіть клас під назвою `Student` із двома екземплярами `student1`, `student2` і призначте дані значення атрибутам зазначених екземплярів. Вивести всі атрибути екземплярів `student1`, `student2` з їхніми значеннями в заданому форматі.
35. Напишіть програму, яка читає два цілих числа, що представляють місяць і день, і друкують сезон для цього місяця і дня
36. Напишіть програму Python для друку зазначеного списку після видалення 0-го, 4-го та 5-го елементів `['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']`
37. Напишіть програму Python для створення та друку списку перших і останніх 5 елементів, де значення є квадратом чисел від 1 до 30 (обидва включені).
38. Напишіть програму Python для створення та друку списку, за винятком перших 5 елементів, де значення є квадратом чисел від 1 до 30 (обидва включені).
39. Напишіть програму Python для вибору випадкового елемента зі списку, набору, словника (значення) і файлу з каталогу.
40. Напишіть програму Python для генерації випадкового символу алфавіту, алфавітного рядка та алфавітного рядка фіксованої довжини.
41. Напишіть програму на Python для створення генератора випадкових чисел із початковим кодом, а також генеруйте значення з плаваючою точкою між 0 і 1, за винятком 1.
42. Напишіть програму на Python для генерації випадкового цілого числа від 0 до 6, за винятком 6, випадкового цілого числа від 5 до 10, за винятком 10, випадкового цілого числа від 0 до 10 із кроком 3 і випадковою датою між двома датами.
43. Напишіть програму Python для перемішування елементів заданого списку.
44. Напишіть програму Python для генерації числа з плаваючою точкою від 0 до 1 включно та генерування випадкового числа з плаваючою точкою в межах певного діапазону (1-10)
45. Напишіть програму на Python, щоб перевірити, чи є функція визначеною користувачем.
46. Напишіть програму для створення текстового віджета за допомогою модуля `tkinter`. Вставте рядок на початку, а потім вставте рядок у поточний текст. Видалити перший і останній символи тексту.
47. Напишіть програму Python, щоб перевірити, чи дане значення є скомпільованим кодом чи ні. Також перевірте, чи дане значення є модулем чи ні.
48. Напишіть програму на Python, щоб створити поверхневу копію заданого списку.
49. Напишіть програму Python для створення глибокої копії заданого списку.
50. Напишіть програму на Python для створення неглибокої копії заданого словника.
51. Напишіть програму Python, щоб отримати поточну адресу пам'яті та довжину в елементах буфера, який використовується для зберігання вмісту масиву, а також знайти розмір буфера пам'яті в бай Напишіть програму Python для додавання елементів із `inerrable` у кінець масиву.
52. Напишіть програму Python для перетворення масиву в масив машинних значень і повернення представлення байтів

Робочу програму навчальної дисципліни (силабус):

Складено доцентом кафедри КМ , к.т.н., Івановським Олексієм Анатолійовичем

Ухвалено кафедрою Конструювання машин (протокол № 13 від 19.07.2022 року)

Погоджено Методичною комісією факультету (протокол № 11 від 29.08.2022 року)