



Основи програмування. Частина 1. Базові конструкції

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 курс, осінній семестр</i>
Обсяг дисципліни	<i>Лекції: 36 год., комп'ютерний практикум: 18 год., лабораторні роботи: 36 год., самостійна робота: 75 год.</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен, модульна контрольна робота, опитування, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на поточний навчальний рік (rozklad.kpi.ua)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., асистент, Погорелов Володимир Володимирович Практичні роботи: к.т.н, асистент, Погорелов Володимир Володимирович</i>
Розміщення курсу	<i>Google classroom. Доступ надається зареєстрованим користувачам</i>

Програма навчальної дисципліни

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

Вивчення дисципліни «Основи програмування. Частина 1. Базові конструкції» спрямована на вивчення теоретичних та методологічних основ побудови програм мовою програмування C#, оволодіння засобами створення програмного забезпечення, отримання практичних навичок розробки програмного забезпечення при вирішенні практичних задач. Набуті знання студенти можуть використовувати під час створення консольних додатків для оброблення текстових чи мультимедійних даних, використання простих структур даних та стандартних файлових форматів і забезпечення командного інтерфейсу користувача.

Метою вивчення дисципліни є ознайомлення студентів з основною термінологією галузі інформаційних технологій, оволодіння теорією програмування для застосовування на практиці; набуття здатностей аналізувати завдання та будувати алгоритми для їх вирішення, реалізовувати базові алгоритми мовою програмування, опанувати сучасні інструменти та технології розроблення програмного забезпечення.

Предметом дисципліни «Основи програмування. Частина 1. Базові конструкції» є вивчення найпростіших основних алгоритмічних та програмних конструкцій для створення обчислювальних програм, а також написання алгоритмів, які мають практичне застосування.

Вивчення дисципліни «Основи програмування. Частина 1. Базові конструкції» формує у здобувачів освіти **загальні (ЗК)** та **фахові компетентності (ФК)**:

ЗК01 Здатність до абстрактного мислення, аналізу та синтезу.

ЗК06 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ФК01 Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК02 Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК03 Здатність розробляти архітектури, модулі та компоненти програмних систем. **ФК07**

Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК08 Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК10 Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК11 Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розроблення програмного забезпечення.

ФК13 Здатність обґрунтовано обирати та освоювати інструментарій з розроблення та супроводження програмного забезпечення.

ФК14 Здатність до алгоритмічного та логічного мислення.

Програмні результати навчання (ПРН) дисципліни «Основи програмування. Частина 1. Базові конструкції» за освітньою програмою:

ПРН01 Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН06 Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН07 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН13 Знати і застосовувати методи розроблення алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПРН15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПРН18 Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

ПРН38 Вміти застосовувати технології програмування для розроблення програмного забезпечення мультимедійних та інформаційно-пошукових систем.

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

Успішному вивченню дисципліни «Основи програмування. Частина 1. Базові конструкції» передують вивчення дисциплін шкільного курсу математики (для аналізу числових даних, які описуються математичними законами) та інформатики (для засобів оброблення та зберігання даних на персональному комп'ютері).

Отримані при засвоєнні дисципліни «Основи програмування. Частина 1. Базові конструкції» теоретичні знання та практичні уміння є необхідними для більшості дисциплін навчального плану та освітньої програми в цілому, зокрема, вивчення дисциплін «Програмування», «Бази даних», «Компоненти програмної інженерії» та інших дисциплін навчального плану підготовки бакаврів за спеціальністю 121 Інженерія програмного забезпечення, а також для успішного проходження переддипломної практики, виконання курсових та дипломних проєктів за спеціальністю 121 Інженерія програмного забезпечення.

3. Зміст навчальної дисципліни

Дисципліна «Основи програмування. Частина 1. Базові конструкції» передбачає вивчення таких тем:

Тема 1. Вступ до мови програмування С#. Структурні елементи програми мовою програмування С#. Основні типи даних у мові програмування С#. Операції та їх пріоритет. Літерали.

Тема 2. Форматування у мові програмування С#. Форматоване введення-виведення даних.

Тема 3. Керуючі оператори: розгалуження і переходи.

Тема 4. Керуючі оператори: цикли, вкладені цикли та цикли з розгалуженням.

Тема 5. Символьне введення-виведення і перевірка на коректність уведених користувачем даних.

Тема 6. Функції. Використання вбудованих функцій мови програмування С# та створення власних.

Тема 7. Масиви та колекції. Символьні рядки і рядкові функції. Структури та інші форми даних.

Тема 8. Введення-виведення. Робота з файлами.

Тема 9. Маніпулювання окремих бітів. Побітова арифметика. Модульна арифметика.

Модульна контрольна робота

Екзамен

4. Навчальні матеріали та ресурси

Базова література:

1. Навчальні матеріали з дисципліни «Основи програмування». Використати для опанування теоретичних знань та практичних умінь з дисципліни. Матеріали знаходяться в Google classroom. Доступ надається зареєстрованим студентам.
2. Керніган Б., Річі Д. Мова програмування С. Друге видання: пер.В. Цибуляк. 2012. 222 с.
3. Коваленко І.В. Програмування мовою С# 6.0. Тернопіль, ТНТУ. 2016. 227с.

Додаткова література:

1. Albahari, J. C# 10 in a Nutshell: The Definitive Reference. O'Reilly Media, 2022. 1058 p.
2. Windows Forms. Программирование на С# [електронний ресурс]. Режим доступу: <http://csharpcoding.org/category/windows-forms/>.
3. Wagner, Bill. Effective C# / Bill Wagner. Addison-Wesley Professional, 2016. 288 p.
4. Deitel, P., Deitel, H. Visual C# How to Program / P. Deitel, H. Deitel. - Pearson, 2016. 1056 p.
5. Price, Mark J. C# 8.0 and .NET Core 3.0 / Mark J. Price. - Packt Publishing, 2019. 820 p.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
-------	-------------------------	--------------------------

Тема 1. Вступ до мови програмування C#.

1	<i>Лекція 1. Структурні елементи програми мовою програмування C#. Основні типи даних у мові програмування C#. Операції та їх пріоритет. Літерали.</i>	<i>Опрацювати теоретичний матеріал першого лекційного заняття. Файл презентації лекційного заняття знаходиться у classroom (посилання наведено вище). Вивчити основні типи даних, які використовуються у мові програмування C#. Визначити їх особливості при явному та неявному перетворенні даних. Завдання на СРС: п. 1.</i>
2	<i>Комп'ютерний практикум 1. Лінійні алгоритми. Типи даних. Написання найпростіших програм мовою програмування C#</i>	<i>Завдання: написати консольні програмні додатки для задач (кількість та їхні постановки вказані в classroom), які подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 2.</i>
3	<i>Лабораторна робота 1. Написання найпростіших програм мовою програмування C#</i>	<i>Завдання: написати консольний додаток для задачі, вказаної в classroom за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 3.</i>

Тема 2. Форматування у мові програмування C#. Форматоване введення-виведення даних.

4	<i>Лекція 2. Особливості форматованого виведення. Кодування даних. Системи ASCII та Unicode</i>	<i>Опанувати теоретичний матеріал лекції, розглянути усі програмні коди, що наведені у презентації лекції (за посиланням до classroom) Завдання на СРС: п. 4.</i>
5	<i>Лекція 3. Методи представлення текстової інформації у консольних застосунках</i>	<i>Опанувати теоретичний матеріал лекції, розглянути усі програмні коди, що наведені у презентації лекції (за посиланням до classroom) Завдання на СРС: п. 5.</i>

Тема 3. Керуючі оператори: розгалуження і переходи

6	<i>Лекція 4. Конструкції мови if, if-else, оператор умовного вибору switch-case. Алгоритми з розгалуженням</i>	<i>Опрацювати теоретичний матеріал другого лекційного заняття (презентація за посиланням до classroom). Розглянути алгоритми, які подані у презентації для другої лекції, запрограмувати відповідні алгоритми. Завдання на СРС: п. 6.</i>
7	<i>Комп'ютерний практикум 2. Створення програмних застосунків з використанням алгоритмів розгалуження</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен</i>

		<i>студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 7.</i>
8	<i>Лабораторна робота 2</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 8.</i>
9	<i>Лабораторна робота 3</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 9.</i>
10	<i>Лабораторна робота 4</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 10.</i>
<i>Тема 4. Керуючі оператори: цикли, вкладені цикли та цикли з розгалуженням</i>		
11	<i>Лекція 5. Керуючі оператори мови програмування C#: циклічні структури алгоритми, які використовують цикли. Програмування алгоритмів з використанням циклів</i>	<i>Опрацювати теоретичний матеріал другого лекційного заняття (презентація за посиланням до classroom). Розглянути алгоритми на базі циклічних структур, які подані у презентації для п'ятої лекції, запрограмувати відповідні алгоритми. Завдання на СРС: п. 11.</i>
12	<i>Лекція 6. Керуючі оператори мови програмування C#: практичні алгоритми, які вимагають циклічної обробки даних. Обчислювальні задачі на циклічні алгоритми</i>	<i>Опрацювати теоретичний матеріал лекції (файл презентації у classroom за посиланнями вище). Виконати компіляцію та запуск відповідних проєктів з програмами. Опрацювати основні алгоритми, які запрограмовані у відповідних проєктах Завдання на СРС: п. 12.</i>
13	<i>Комп'ютерний практикум 3. Створення програмних застосунків з використанням циклічних алгоритмів</i>	<i>Завдання: виконати завдання за індивідуальним варіантом для кожного студента. Постановки завдань наведені у комп'ютерному практикуму (за посиланням до classroom). Запрограмувати циклічні алгоритми для розв'язку поставлених задач та отримані результати роботи програм оформити у звіт та показати викладачеві. Завдання на СРС: п. 13.</i>

14	Лабораторна робота 5	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 14.
15	Лабораторна робота 6	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 15.
<i>Тема 5. Символьне введення-виведення і перевірка на коректність уведених користувачем даних</i>		
16	Лекція 7. Методи забезпечення коректності та перевірки адекватності уведених користувачем даних у консольних додатках мовою програмування C#	Опанувати основні принципи та методи верифікації уведених даних користувачем програмного застосунку. Скопіювати усі програми, які наведені у презентації (файл презентації лекції за посиланням до classroom). Завдання на СРС: п. 16.
<i>Тема 6. Функції. Використання вбудованих функцій мови програмування C# та створення власних</i>		
17	Лекція 8. Актуальність та методи створення функцій у власних програмах. Приклади практичних задач, які вимагають використання власних функцій у розроблюваних проєктах. Рекурсивні функції та робота з ними.	Опанувати теоретичний матеріал щодо створення функцій. Опрацювати теоретичний лекційний матеріал (за посиланням до classroom), виконати компіляцію, запуск та тестування розглянутих у лекції програмних кодів проєктів. Завдання на СРС: п. 17.
18	Комп'ютерний практикум 4. Створення власних функцій для оброблення даних різного походження	Завдання: виконати завдання комп'ютерного практикуму за варіантом кожному студенту. Постановки завдань комп'ютерного практикуму наведені у відповідному файлі (за посиланням до classroom). Отримані результати виконання практикуму занести до звіту та представити його викладачеві. Завдання на СРС: п. 18.
19	Лабораторна робота 7	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен

		студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 19.
20	Лабораторна робота 8	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 20.
<i>Тема 7. Масиви та колекції. Символьні рядки і рядкові функції. Структури та інші форми даних.</i>		
21	Лекція 9. Масиви та колекції, найпростіші алгоритми для роботи з ними. Символьні рядки і рядкові функції.	Опрацювати теоретичний матеріал з масивами та колекціями, зрозуміти принципи їх роботи. Завдання на СРС: п. 21.
22	Комп'ютерний практикум 5. Створення програмних застосунків зі збереженням даних у пам'яті. Використання одно- та багатовимірних статичних масивів, динамічних масивів, зубчастих масивів	Завдання: написати програмні застосунки згідно з варіантом для кожного студента з індивідуальними завданнями (постановки задач за посиланням в classroom). Навчитись обробляти графічну інформацію, яка зберігається у вигляді масивів. Завдання на СРС: п. 22.
23	Лабораторна робота 9	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 23.
24	Лекція 10. Структури даних та інші форми подання даних (частина 1)	Опрацювати теоретичний матеріал зі статичними структурами даних, зрозуміти принципи та доцільність використання складних типів даних (файл з матеріалами лекційного заняття за посиланням до classroom). Завдання на СРС: п. 24.
25	Лекція 11. Структури даних та інші форми подання даних (частина 2)	Опрацювати теоретичний матеріал зі динамічними структурами даних, зрозуміти принципи та доцільність використання складних типів даних (файл з матеріалами лекційного заняття за посиланням до classroom). Завдання на СРС: п. 25.
26	Комп'ютерний практикум 6. Розроблення програмних застосунків для оброблення текстової та графічної інформації з використанням масивів, рядків та структур даних	Завдання: написати програми для оброблення текстової інформації, створення простої системи розпізнавання та визначення правдоподібності текстових даних на базі машинного навчання. Завдання на СРС: п. 26.

27	Лабораторна робота 10	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 27.
28	Лабораторна робота 11	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 28.
<i>Тема 8. Файлове введення-виведення.</i>		
29	Лекція 12. Класи мови програмування C# для роботи з файлами.	Опрацювати теоретичний матеріал з класами StreamReader та StreamWriter. Навчитись виконувати запис за зчитування даних текстового файлу. Матеріали лекції знаходяться в classroom (за посиланням) Завдання на СРС: п. 29.
30	Лекція 13. Режими оброблення текстової інформації у файлах	Опрацювати теоретичний матеріал лекційного заняття (файл за посиланням до classroom). Скопіювати усі програмні коди, які наведені у лекційному занятті. Завдання на СРС: п. 30.
31	Комп'ютерний практикум 7. Оброблення текстової інформації. Робота з файлами	Завдання: розробити програми для індивідуальних завдань (за варіантом для кожного студента), що вимагають запису, зчитування даних з файлів.. Завдання на СРС: п. 31.
32	Лабораторна робота 12	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 32.
33	Лабораторна робота 13	Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 33.
<i>Тема 9. Маніпулювання бітами. Модулярна арифметика.</i>		

34	<i>Лекція 14. Формування бітової інформації. Побітові операції. Робота з множинами</i>	<i>Опрацювати теоретичний матеріал лекційного завдання (файл за посиланням до classroom). Розглянути та протестувати описані у роботі алгоритми. Завдання на СРС: п. 34.</i>
35	<i>Комп'ютерний практикум 8. Написання консольних програмних застосунків для роботи з бітовими операціями, а також з множинами</i>	<i>Завдання: написати програмні застосунки згідно з варіантом для кожного студента з індивідуальними завданнями (постановки задач за посиланням в classroom). Завдання на СРС: п. 35.</i>
36	<i>Лабораторна робота 14</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 36.</i>
37	<i>Лабораторна робота 15</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 37.</i>
38	<i>Лекція 15. Базові математичні основи модулярної арифметики.</i>	<i>Розглянути лекційний матеріал, опрацювати теоретичні відомості, що базуються на найпростіших алгоритмах модулярної арифметики. Завдання на СРС: п. 38.</i>
39	<i>Лекція 16. Практичне застосування модулярної арифметики. Елементи криптографії (частина 1)</i>	<i>Опанувати математичні основи модулярної арифметики, які наведено у лекційному матеріалі (файл додається до classroom за посиланням). Завдання на СРС: п. 39.</i>
40	<i>Лекція 17. Практичне застосування модулярної арифметики. Елементи криптографії (частина 2)</i>	<i>Проаналізувати основні алгоритми практичного застосування модулярної арифметики, розглянути відповідні програмні коди, які наведені у лекційному матеріалі (посилання на файл за посиланням до classroom). Завдання на СРС: п. 40.</i>
41	<i>Комп'ютерний практикум 9. Модулярна арифметика та її практичне застосування</i>	<i>Завдання: виконати завдання комп'ютерного практикуму за варіантом у списку для кожного студента. Кожен варіант містить індивідуальні завдання, постановки яких наведені у самому практикуму (за посиланням до classroom). Завдання на СРС: п. 41.</i>
42	<i>Лабораторна робота 16</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен</i>

		<i>студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 42.</i>
43	<i>Лабораторна робота 17</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 43.</i>
44	<i>Лабораторна робота 18</i>	<i>Завдання: виконати індивідуальні завдання за варіантом, який наведений у таблиці. Кількість завдань та їхні постановки вказані в classroom, подані за варіантами (кожен студент обирає власний варіант за списком в межах групи). Завдання на СРС: п. 44.</i>
<i>Модульна контрольна робота</i>		

6. Самостійна робота студента/аспіранта

Дисципліна «Основи програмування. Частина 1. Базові конструкції» ґрунтується на самостійній підготовці до аудиторних занять на теоретичні та практичні теми.

<i>№ з/п</i>	<i>Назва теми, що виноситься на самостійне опрацювання</i>	<i>Кількість годин</i>	<i>Література</i>
1	<i>Підготовка до лекції 1</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
2	<i>Підготовка до комп'ютерного практикуму 1</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
3	<i>Підготовка до лабораторної роботи 1</i>	1	<i>Основна: 1-3</i>
4	<i>Підготовка до лекції 2</i>	1	<i>Основна: 1-3 Додаткова: 1; 4</i>
5	<i>Підготовка до лекції 3</i>	1	<i>Основна: 1-3 Додаткова: 1; 3</i>
6	<i>Підготовка до лекції 4</i>	1	<i>Основна: 1-3</i>
7	<i>Підготовка до комп'ютерного практикуму 2</i>	1	<i>Основна: 1-3</i>
8	<i>Підготовка до лабораторної роботи 2</i>	1	<i>Основна: 1-3</i>
9	<i>Підготовка до лабораторної роботи 3</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
10	<i>Підготовка до лабораторної роботи 4</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
11	<i>Підготовка до лекції 5</i>	1	<i>Основна: 1-3 Додаткова: 1; 4</i>

12	<i>Підготовка до лекції 6</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
13	<i>Підготовка до комп'ютерного практикуму 3</i>	1	<i>Основна: 1-3 Додаткова: 1; 4</i>
14	<i>Підготовка до лабораторної роботи 5</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
15	<i>Підготовка до лабораторної роботи 6</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
16	<i>Підготовка до лекції 7</i>	1	<i>Основна: 1-3</i>
17	<i>Підготовка до лекції 8</i>	1	<i>Основна: 1-3</i>
18	<i>Підготовка до комп'ютерного практикуму 4</i>	1	<i>Основна: 1-3 Додаткова: 1; 4</i>
19	<i>Підготовка до лабораторної роботи 7</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
20	<i>Підготовка до лабораторної роботи 8</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
21	<i>Підготовка до лекції 9</i>	1	<i>Основна: 1-3 Додаткова: 1; 2</i>
22	<i>Підготовка до комп'ютерного практикуму 5</i>	1	<i>Основна: 1-3 Додаткова: 2</i>
23	<i>Підготовка до лабораторної роботи 9</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
24	<i>Підготовка до лекції 10</i>	1	<i>Основна: 1-3</i>
25	<i>Підготовка до лекції 11</i>	1	<i>Основна: 1-3</i>
26	<i>Підготовка до комп'ютерного практикуму 6</i>	1	<i>Основна: 1-3</i>
27	<i>Підготовка до лабораторної роботи 10</i>	1	<i>Основна: 1-3 Додаткова: 1</i>
28	<i>Підготовка до лабораторної роботи 11</i>	1	<i>Основна: 1-3</i>
29	<i>Підготовка до лекції 12</i>	1	<i>Основна: 1-3</i>
30	<i>Підготовка до лекції 13</i>	1	<i>Основна: 1-3</i>
31	<i>Підготовка до комп'ютерного практикуму 7</i>	1	<i>Основна: 1-3</i>
32	<i>Підготовка до лабораторної роботи 12</i>	1	<i>Основна: 1-3</i>
33	<i>Підготовка до лабораторної роботи 13</i>	1	<i>Основна: 1-3</i>
34	<i>Підготовка до лекції 14</i>	1	<i>Основна: 1-3</i>
35	<i>Підготовка до комп'ютерного практикуму 8</i>	1	<i>Основна: 1-3</i>
36	<i>Підготовка до лабораторної роботи 14</i>	1	<i>Основна: 1-3</i>

37	Підготовка до лабораторної роботи 15	1	Основна: 1-3
38	Підготовка до лекції 15	1	Основна: 1-3
39	Підготовка до лекції 16	1	Основна: 1-3
40	Підготовка до лекції 17	1	Основна: 1-3 Додаткова: 2; 3
41	Підготовка до комп'ютерного практикуму 9	1	Основна: 1-3 Додаткова: 3
42	Підготовка до лабораторної роботи 16	1	Основна: 1-3
43	Підготовка до лабораторної роботи 17	1	Основна: 1-3
44	Підготовка до лабораторної роботи 18	2	Основна: 1-3
45	Підготовка до екзамену	30	Основна: 1-3 Додаткова: 1-5

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування занять. Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також презентація / публічний виступ та участь у обговореннях та доповнення на семінарах оцінюватимуться під час аудиторних занять. Для активної участі у роботі семінару студент готується за рекомендованою викладачем до певного семінарського заняття літературою. Участь у роботі семінару також передбачає підготування доповідей та співдоповідей у межах усіх занять.

Пропущені контрольні заходи оцінювання. Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи. Детальніше за посиланням: <https://kpi.ua/files/n3277.pdf>.

Процедура оскарження результатів контрольних заходів оцінювання. Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами. Студенти мають право аргументовано оскаржити результати контрольних заходів, пояснивши з яким критерієм не погоджуються відповідно до оціночного. Календарний контроль проводиться з метою підвищення якості навчання студентів та моніторингу виконання студентом вимог силабусу.

Академічна доброчесність. Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Норми етичної поведінки. Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Інклюзивне навчання. Засвоєння знань та умінь в ході вивчення дисципліни «Науково-дослідна діяльність у комп'ютерній інженерії» може бути доступним для більшості осіб з особливими

освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

Навчання іноземною мовою. У ході виконання завдань студентам може бути рекомендовано звернутися до англomовних джерел. Призначення заохочувальних та штрафних балів Відповідно до Положення про систему оцінювання результатів навчання сума всіх заохочувальних балів не може перевищувати 10% рейтингової шкали оцінювання.

Всі студенти повинні відвідувати лекційні та практичні заняття, на яких потрібно активно працювати над засвоєнням навчального матеріалу. За об'єктивних причин (наприклад - хвороба, міжнародне стажування) навчання може відбуватись в он-лайн формі індивідуально за погодженням із керівником курсу.

Політика щодо дедлайнів та перескладання:

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перескладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності:

Усі письмові роботи перевіряються на наявність плагіату і допускаються до захисту із коректними текстовими запозиченнями не більше 20%. Списування під час контрольних робіт заборонені (в т. ч. із використанням мобільних пристроїв).

Заохочувальні бали нараховуються за:

- дострокове виконання комп'ютерного практикуму з відповідним захистом звіту з результатами виконання даного практикуму (максимальна кількість балів – 1 бал за кожен практикум).
- творчий підхід у виконанні робіт комп'ютерного практикуму (максимальна кількість балів за кожну роботу – 1 бал).

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

Протягом семестру студенти виконують 9 комп'ютерних практикумів. Максимальна кількість балів за кожний комп'ютерний практикум: 4 бали.

Бали нараховуються за:

- якість виконання комп'ютерного практикуму: 0–2 бали;
- відповідь під час захисту комп'ютерного практикуму: 0–2 бали;

Критерії оцінювання якості виконання роботи:

- 2 бали – робота виконана якісно, в повному обсязі;
- 1–1,5 бали – робота виконана якісно, в повному обсязі, але має недоліки;
- 0–0,5 бали – робота виконана не в повному обсязі, або містить суттєві помилки або відсутня.

Критерії оцінювання відповіді:

- 2 бали – відповідь повна, добре аргументована;
- 1–1,5 бали – в цілому відповідь вірна, але має недоліки або незначні помилки;
- 0–0,5 бали – у відповіді є суттєві помилки або вона відсутня;

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:
4 бали × 9 комп'ютерних практикумів (комп'ютерні практикуми) = 36 балів.

Двічі на семестр виконується **модульна перевірка знань (модульна контрольна робота)** у вигляді практичних знань з дисципліни. Студентами виконується написання програм за пройденими темами. Максимальна кількість балів за кожну з двох робіт: 7 балів. Загальна оцінка за дві атестації не перевищує 14 балів.

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 50 % від максимальної кількості балів, яку може отримати студент під час першої атестації.

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 50 % від максимальної кількості балів, яку може отримати студент до другої атестації.

Семестровий контроль: екзамен

Умови допуску до семестрового контролю:

При семестровому рейтингу (r_c) не менше 30 балів (60% від максимуму) та зарахуванні усіх видів робіт, студент допускається до екзамену.

Завдання на **екзаменаційну роботу** складається з 5 питань – 2 теоретичних та 3 практичних. Відповідь на кожне теоретичне запитання оцінюється 10 балами максимум, а відповідь на практичне запитання оцінюється 10 балів максимум.

Критерії оцінювання кожного теоретичного запитання контрольної роботи:

10 бали – відповідь вірна, повна, добре аргументована;

7-9 бали – відповідь вірна, розгорнута, але не дуже добре аргументована;

3-6 бали – відповідь не повна та неаргументована та має суттєві помилки, які не дають можливість отримати вірний результат до задачі / завдання;

0-2 бали – відповідь не вірна, відповіді немає.

Критерії оцінювання практичного запитання контрольної роботи:

10- балів – відповідь вірна, програмні розрахунки виконані у повному обсязі;

8-9 балів – відповідь вірна, але програмні розрахунки не завжди дають вірний результат;

5-7 балів – у відповіді є незначні помилки, які суттєво впливають на результати програмних розрахунків;

0-4 бали – у відповіді є суттєві помилки, немає відповіді або відповідь невірна.

Максимальна кількість балів за екзаменаційну роботу:

10 бали × 2 теоретичні запитання + 10 балів × 3 практичних запитання = 50 балів.

Семестрова складова рейтингової шкали $R_c = 50$ балів, вона визначається як сума балів, отриманих за виконання та захист робіт комп'ютерного практикуму, та балів, отриманих за модульну контрольну роботу.

$R_c = R_{\text{комп. прак.}} + R_{\text{мкр}} = 36 \text{ бали} + 14 \text{ балів} = 50 \text{ балів.}$

Склад та критерії оцінювання відповіді на екзамені:

Екзаменаційна складова рейтингової шкали $R_E = 50$ балів.

Рейтингова шкала з дисципліни дорівнює: $R = R_c + R_E = 50 \text{ балів} + 50 \text{ балів} = 100 \text{ балів}$

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо

Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., асистент Погорелов В.В.

Ухвалено кафедрою ПЗКС (протокол № 13 від 22.06.2022 р.)

Погоджено Методичною комісією факультету прикладної математики (протокол № 9 від 24.06.2022 р.)

Перелік і основних алгоритмів до іспиту з дисципліни

- 1. Структура простої програми мовою програмування C#.*
- 2. Системи числення (двійкова, вісімкова, десяткова та шістанцяткова). Перехід між даними системами числення. Додавання та віднімання чисел у даних системах числення.*
- 3. Типи даних, які використовуються при написанні програм мовою програмування C#. Характеристики даних типів (загальний обсяг пам'яті в ПК та діапазони значень, методи їх отримання).*
- 4. Алфавіт і синтаксис мови. Ідентифікатори / лексеми, константи. Зарезервовані імена мови програмування C#. Лексика мови. Змінні і константи. Вирази і операції.*
- 5. Оператори введення-виведення. Форматований ввід та вивід даних у C#.*
- 6. Правила запису виразів і операцій.*
- 7. Таблиця кодових символів ASCII. Отримання коду символа та отримання символа за його кодом.*
- 8. Клас математичних функцій Math (мова C#) та його використання.*
- 9. Основні алгоритмічні конструкції: лінійні, розгалужуються, циклічні. Приклади завдань, які вирішуються з використанням застосуванням даних конструкцій та їх комбінацій.*
- 10. Логічні операції, їхні властивості та робота з цими операціями. Таблиці істинності основних логічних операцій. Складання складних логічних умов.*
- 11. Оператори розгалуження. Синтаксис та приклади використання у конкретних задачах.*
- 12. Циклічні конструкції. Цикли з передумовою та післяумовою. Синтаксис та приклади використання.*
- 13. Масиви. Синтаксис оголошення масиву. Операції над масивами. Стандартні функції для роботи з масивами. Введення і виведення масивів.*
- 14. Багатовимірні масиви. Синтаксис та робота з багатовимірними масивами.*
- 15. Стандартні колекції мови програмування C# (наприклад, List).*
- 16. Рядки і множини. Синтаксис оголошення рядків і множин. Операції над рядками і множинами. Стандартні функції і процедури для роботи з рядками.*
- 17. Структуровані типи даних: рядки і множини. Створення власних складних типів даних (структури).*
- 18. Процедури і функції. Поняття підпрограми. Синтаксис оголошення процедур і функцій. Стандартні процедури і функції мови програмування.*
- 19. Формальні і фактичні параметри. Процедури з параметрами, опис процедур.*
- 20. Функції: способи організації та опис. Виклик функцій, рекурсія.*
- 21. Програмування рекурсивних алгоритмів.*
- 22. Операції з бітами, функції для роботи з бітами. Побітовий зсув.*
- 23. Організація введення-виведення даних. Робота з файлами.*

Основні алгоритми, які виносяться на іспит з дисципліни «Основи програмування 1. Базові конструкції»

- 1. Алгоритм сортування бульбашкою. Сортування списку за алфавітом.*
- 2. Алгоритми обміну значень двох змінних.*
- 3. Алгоритм пошуку мінімуму / максимуму у масивах даних.*
- 4. Алгоритм множення матриць на рівні програмного коду.*
- 5. Алгоритм Евкліда для знаходження НСД двох натуральних чисел. Використання отриманого результату для визначення НСК цих чисел.*
- 6. Алгоритм знаходження суми / добутку елементів числової послідовності, яка задана загальним її членом.*
- 7. Алгоритм знаходження суми членів функціонального ряду, який заданий загальним її членом, з заданою точністю.*
- 8. Алгоритми матричної фільтрації зображень*

9. *Стеганографічна обробка зображень. Вбудовування бінарного водяного знаку у вхідне зображення. Вилучення водяного знаку з зображення (відновлення вхідного зображення).*
10. *Злиття двох зображень.*
11. *Алгоритм Дейкстри для пошуку оптимального шляху.*
12. *Алгоритм Цезаря для шифрування даних.*
13. *Алгоритм швидкого піднесення до степеня числа за модулем.*
14. *Алгоритм пошуку оберненого за модулем елемента (розширений алгоритм Евкліда)*
15. *Алгоритм шифрування і розшифрування RSA.*
16. *Алгоритм обміну ключами Діффі-Хеллмана.*

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., ас. Погорелов В.В.

Ухвалено кафедрою ПЗКС (протокол № 12 від 26.04.2023 р.)

Погоджено Методичною комісією факультету прикладної математики (протокол № 10 від 26.05.2023 р.)