



ОСНОВИ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

1. Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем і веб-технологій
Статус дисципліни	Нормативна
Форма навчання	Заочна
Рік підготовки, семестр	I курс, осінній/весінній семестр
Обсяг дисципліни	330 годин / 11 кредитів ЄКТС 5,5 кредитів/165 годин – перший кредитний модуль «Основи програмування – 1. Базові конструкції»; 5,5 кредитів/165 годин – другий кредитний модуль «Основи програмування – 2. Модульне програмування»;
Семестровий контроль/ контрольні заходи	Екзамен, мкр
Розклад занять	http://rozklad.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	<u>Лектор:</u> к.т.н., доцент Шпурик Вадим Вадимович <u>Практичні заняття:</u> к.т.н., доцент Шпурик Вадим Вадимович <u>Лабораторні роботи:</u> к.т.н., доцент Шпурик Вадим Вадимович
Розміщення курсу	Кампус – campus.kpi.ua http://route.ho.ua/kpi/c1/ http://route.ho.ua/kpi/c2/

2. Програма навчальної дисципліни

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

Майбутньому фахівцеві варто вивчати дисципліну «Основи програмування» тому, що вона є базовим курсом для всіх тих, хто хоче вивчити мови програмування, зокрема, C і C++, та стати досвідченим розробником програмного забезпечення. Курс містить також спеціальні розділи, присвячені життєвому циклу програмних засобів сучасної IT-індустрії, проблеми помилок в програмах і методах верифікації програмного забезпечення, стилю програмування. Після проходження лабораторного практикуму студент буде здатний писати програми середньої складності на мовах C, C++. Процес побудований таким чином, що слухачі навчаються не тільки

писати код і розуміти код, написаний іншими програмістами, але і вирішувати проблеми, з якими неминуче доведеться зіткнутися при створенні програмного забезпечення. Слід окремо підкреслити, що хоча цей курс викладається українською мовою, в цьому курсі ви паралельно вивчаєте і вдосконалюєте свою англійську мову і специфічну термінологію ІТ-галузі. Це допоможе як з розумінням програмування (більшість документації по мовам програмування написана саме англійською мовою), так і в подальшому з пошуком роботи.

Метою опанування дисципліни є освоєння фундаментальних основ програмування; ознайомлення з усіма основними термінами і поняттями; здобуття практичних навичок розробки алгоритмів; отримання реального досвіду розробки; навчитися писати доволі складні програми; познайомитися з передовими мовами програмування і напрямками програмування; чітко розуміти, який напрямок вам найцікавіше; отримати матеріали та інструкції для подальшого розвитку.

Предметом дисципліни є вивчення фундаментальних принципів сучасного процедурного та об'єктно-орієнтованого програмування на базі мов системного програмування C, C++.

Програмні результати навчання

Дисципліна сприяє формуванню у студентів певних компетентностей, тобто здатностей до виконання певних завдань самостійно. Компетентності формуються як фахові, так і загальні.

Загальні компетентності:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Фахові компетентності:

- здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення;
- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;
- здатність розробляти архітектури, модулі та компоненти програмних систем;
- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- здатність накопичувати, обробляти та систематизувати професійні знання щодо створення тестування і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя;
- здатність реалізувати фази та ітерації життєвого циклу програмних систем інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення;
- здатність обґрунтовано обирати та освоювати інструментарій з розробки тестування та супроводження програмного забезпечення;
- здатність до алгоритмічного та логічного мислення;
- володіти скриптовими та декларативними мовами програмування.

Згідно з вимогами програми навчальної дисципліни після засвоєння навчальної дисципліни студенти мають продемонструвати певні результати навчання, тобто набуті ними знання й уміння описані далі.

- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення;

- уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;
- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;
- знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

2. Пререквізити та постреквізити дисципліни

Вивчення дисципліни «*Основи програмування*» проходить паралельно з вивченням студентами основоположних дисциплін за спеціальністю, тому розміщується дисципліна у структурно-логічній схемі паралельно з нормативними освітніми компонентами циклу загальної підготовки та з дисциплінами циклу професійної підготовки, такими як: «*Алгоритми та структури даних*», «*Основи комп'ютерних систем і мереж*» та «*Компоненти програмної інженерії*».

Дисципліна «*Основи програмування*» забезпечує вивчення таких дисциплін, як: «*Об'єктно-орієнтований аналіз та конструювання програмних систем*», «*Основи розробки трансляторів*» і «*Бази даних*», а також, забезпечує написання курсової роботи за навчальною дисципліною «*Основи програмування*» з циклу професійної підготовки «*Курсова робота з основ програмування*».

3. Зміст навчальної дисципліни

Навчальна дисципліна складається з двох кредитних модулів: «*Основи програмування – 1. Базові конструкції*» та «*Основи програмування – 2. Модульне програмування*».

3.1. Перший кредитний модуль

Розділ 1. Введення в мову C: основні поняття

Розділ 2. Дані та конструкції мови C

Тема – 1. Дані

Тема – 2. Керуючі конструкції. Оператори управління

Тема – 3. Введення та виведення даних

Розділ 3. Конструкції та складові мови C

Тема – 1. Функції

Тема – 2. Вказівники: основні відомості, принцип роботи, використання

Тема – 3. Вказівники та масиви

Тема – 4. Функції та масиви

Тема – 5. Символьні рядки та рядкові функції

Розділ 4. Організація програми, взаємодія з файлами мовою C

Розділ 5. Додаткові відомості мови C

Тема – 1. Структури даних

Тема – 2. Додаткові можливості використання вказівників

3.2. Другий кредитний модуль

Розділ 1. Додаткові відомості мови C. Структури даних

Розділ 2. Введення в мову C++: основні поняття та конструкції

Тема – 1. Вступ до C++: основні поняття

Тема – 2. Вступ до C++: конструкції мови

Тема – 3. Масиви об'єктів, використання вказівників і посилання

Розділ 3. Конструкції мови C++

Тема – 1. Використання переважених конструкцій

Тема – 2. Використання наслідування (успадкування)

Розділ 4. Введення та виведення даних у C++. Робота з файлами

Тема – 1. Введення та виведення даних

Тема – 2. Файлове введення та виведення даних

Розділ 5. Віртуальні функції, шаблони та виключення, RTTI

Тема – 1. Поліморфізм і віртуальні функції в програмах

Тема – 2. Шаблони та обробка виключень

Тема – 3. Типи даних і дії з ними

Розділ 6. Простори імен і шаблони проектування

Тема – 1. Простори імен і перетворення типів

Тема – 2. Шаблони проектування

Розділ 7. Додаткові відомості мови C++. Структури даних, хеш-таблиці

4. Навчальні матеріали та ресурси

Для кращого засвоєння дисципліни та виконання лабораторних робіт і самостійної роботи студентам надаються список рекомендованої літератури та навчальні матеріали, розміщені на Інтернет і університетських ресурсах.

Базові ресурси та література:

1. Конспект лекцій. 1 семестр – [Електронний ресурс] – режим доступу: <http://route.ho.ua/kpi/c1/>

2. Конспект лекцій. 2 семестр – [Електронний ресурс] – режим доступу: <http://route.ho.ua/kpi/c2/>

3. Язык программирования C (Си) 2-е издание Брайан У. Керниган, Деннис М. Ритчи – [Електронний ресурс] – режим доступу: <http://www.williamspublishing.com/Books/978-5-8459-0891-9.html#ogl>

4. C: полное руководство, классическое издание Герберт Шилдт. – [Електронний ресурс] – режим доступу: <http://www.williamspublishing.com/Books/978-5-8459-1102-5.html#ogl>

5. Программирование на языке C (Си) 3-е издание Стефан Кочан. – [Електронний ресурс] – режим доступу: <http://www.williamspublishing.com/Books/5-8459-1088-9.html>

6. Перри Г., Миллер Д. Программирование на C для начинающих М.: Эксмо, 2015. — 369 с. — ISBN 978-5-699-75873-9 3-е издание.

Додаткова література:

1. Head First C by David Griffiths and Dawn Griffiths Copyright © 2012 David Griffiths and Dawn Griffiths. All rights reserved. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

2. Подбельский В. В., Фомин С. С. Курс программирования на языке Си: учебник. – М.: ДМК Пресс, 2012. – 384 с.

3. Шпак З.Я. Програмування мовою C: Навчальний посібник Львів: Оріяна-Нова, 2006. - 432 с.

Майк МакГрат. Программирование на C для начинающих. пер. с англ. М. А. Райтмана. 4-е изд 2016.

3. Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна складається з двох кредитних модулів, які охоплюють 11 кредитів або 330 годин. За порядком слідування та розподілом годин кредитні модулі розміщуються на першому курсі в осінньому та весінньому семестрах.

В осінньому семестрі першого курсу передбачено перший кредитний модуль – «Основи програмування – 1. Базові конструкції», на який відводиться 5,5 кредитів або 165 годин. Другий кредитний модуль – «Основи програмування – 2. Модульне програмування», у розмірі 5,5 кредитів або 165 годин розміщується у весінньому семестрі першого року навчання.

Перший та другий кредитні модулі дисципліни включають в себе проведення лекційних занять (6 годин в першому семестрі та 4 години в другому), практичних занять (комп'ютерних

практикумів) – 6 годин в другому семестрі, та лабораторних робіт у розмірі 6 годин у першому та 2 години в другому семестрах.

5.1. Календарний план запланованих заходів на 1 семестр - перший кредитний модуль - «Основи програмування – 1. Базові конструкції»

Термін виконання (тиждень)	Види робіт за розділами та темами
<i>Розділ 1. Введення в мову C: основні поняття</i>	
1	Лекція 1.1. Вступ до мови C. Встановлення та знайомство з компілятором GCC. Структура простої програми. Виконання та компіляція першої програми Лабораторна робота 1.1. Знайомство з компілятором GCC: виконання та компіляція першої програми СРС 1. Встановлення компілятора GCC в середовищі Cygwin і MinGW
<i>Розділ 2. Дані та конструкції мови C</i>	
<i>Тема – 1. Дані мови</i>	
2	Лекція 1.2. Дані: символічні рядки, оператори, вирази та конструкції Лабораторна робота 1.2. Дані мови C: символічні рядки, оператори, вирази та твердження СРС 2. Можливості використання уявних типів даних
<i>Тема – 2. Керуючі конструкції. Оператори управління</i>	
3	Лекція 1.3. Оператори управління: поняття «цикли» Лабораторна робота 1.3. Робота з циклами СРС 3. Випадки використання циклів з перед- та післяумовами (do і do-while)
4	СРС 4. Оператори управління: розгалуження та переходи. Робота з вкладеними циклами. Оптимізація вкладених циклів з метою підвищення ефективності коду
5	СРС 5. Оператори управління: цикли. Продовження. Робота з вкладеними циклами. Продовження. Використання операторів break і continue при роботі з циклами
<i>Тема – 3. Введення та виведення даних</i>	
6	Лекція 1.4. Символьне введення та виведення даних Лабораторна робота 1.4. Символьне введення та виведення даних СРС 6. Різниця між getchar(), gets() і scanf(), можливості їх комбінування з printf()
<i>Розділ 3. Конструкції та складові мови C</i>	
<i>Тема – 1. Функції</i>	
7	Лекція 2.1. Функції. Частина перша: створення та використання простих функцій Лабораторна робота 2.1. Функції. Частина перша: використання простих функцій СРС 7. Робота функції return: основні поняття, виключення та помилки
8	СРС 8. Функції. Частина друга: створення та використання власних функцій. Випадки, в яких використання рекурсії є обов'язковим
9	Лекція 2.2. Функції. Частина друга. Продовження Лабораторна робота 2.2. Функції. Частина друга: використання власних функцій. Продовження СРС 9. Робота з даними типу enum
<i>Тема – 2. Вказівники: основні відомості, принцип роботи, використання</i>	
10	Лекція 3.1. Вказівники. Частина перша: основні поняття та відомості Лабораторна робота 3.1. Вказівники. Частина перша: основні поняття та відомості СРС 10. Одно-, дво- та багатовимірні масиви: основні відомості
11	СРС 11. Вказівники. Частина друга: використання вказівників для роботи з масивами. Аналіз проблем і повідомлень компілятора, що виникають при використанні неініційованих вказівників

Термін виконання (тиждень)	Види робіт за розділами та темами
	<i>Тема – 3. Вказівники та масиви</i>
12	<i>Лекція 3.2. Вказівники. Частина третя: використання вказівників у різних типах сортувань</i> <i>Лабораторна робота 3.2. Частина третя: використання вказівників у різних типах сортувань</i> <i>СРС 12. Вимірювання часу, який витрачається програмою при використанні різних типів сортувань</i>
	<i>Тема – 4. Функції та масиви</i>
13	<i>Лекція 4.1. Функції та масиви</i> <i>Лабораторна робота 4.1. Функції та масиви</i> <i>СРС 13. Які бувають помилки та причини їх виникнення при використанні складеного літералу</i>
	<i>Тема – 5. Символьні рядки та рядкові функції</i>
14	<i>Лекція 4.2. Символьні рядки та рядкові функції</i> <i>Лабораторна робота 4.2. Символьні рядки та рядкові функції</i> <i>СРС 14. Використання UTF-8 зокрема символів кириличної абетки у програмі</i>
<i>Розділ 4. Організація програми, взаємодія з файлами мовою C</i>	
15	<i>Лекція 5.1. Організація програми: масиви, абстрактні дані. Інтерфейс</i> <i>Лабораторна робота 5.1. Організація програми: масиви, абстрактні дані. Інтерфейс</i> <i>СРС 15. Ознаки «хорошого» інтерфейсу</i>
16	<i>Лекція 5.2. Взаємодія з файлами, файлове введення та виведення</i> <i>Лабораторна робота 5.2. Файлове введення та виведення даних</i> <i>СРС 16. Порівняння двійкового та текстового режимів</i>
<i>Розділ 5. Додаткові відомості мови C</i>	
	<i>Тема – 1. Додаткові можливості використання вказівників</i>
17	<i>Лекція 6.1. Розширене використання вказівників</i> <i>Лабораторна робота 6. Файлове введення та виведення даних. Продовження</i> <i>СРС 17. Використання конструкцій “вказівник на вказівник”. Коректне розіменування вказівників</i>
	<i>Тема – 2. Структури даних</i>
18	<i>Лекція 6.2. Структури даних, об’єднання та перелік</i> <i>СРС 18. Масиви структур: використання та застосування</i> <i>Проведення модульної контрольної роботи</i>

5.2. Календарний план запланованих заходів на 2 семестр - другий кредитний модуль - «Основи програмування – 2. Модульне програмування»

Термін виконання (тиждень)	Види робіт за розділами та темами
<i>Розділ 1. Додаткові відомості мови C. Структури даних</i>	
1	<i>СРС 1. Прості абстрактні структури даних. Структури даних на прикладах: масиви та списки, робота з деревами, хеш-структури</i>
2	<i>СРС 2. Випадки «тупикових» ситуацій при роботі з деревами</i>

Термін виконання (тиждень)	Види робіт за розділами та темами
<i>Розділ 2. Введення в мову C++: основні поняття та конструкції</i>	
	<i>Тема – 1. Вступ до C++: основні поняття</i>
3	Лекція 1.1. Вступ до C++ Практичне заняття 1.1. Введення та виведення даних, приведення типів, конструктори та деструктори класів СРС 3. Особливості використання оператора “<<” з різними типами даних
	<i>Тема – 2. Вступ до C++: основні конструкції</i>
4	Лекція 1.2. Вступ до C++. Продовження Практичне заняття 1.2. Основні конструкції: класи, об’єкти та функції СРС 4. Відмінність у використанні struct і class
	<i>Тема – 3. Масиви та використання вказівників</i>
5	Лекція 1.3. Масиви, вказівники та посилання Практичне заняття 1.3. Робота з масивами та їх елементами: виділення пам’яті та контроль границь СРС 5. Безпечні масиви. Обробка ситуацій виходу за межі масиву
<i>Розділ 3. Конструкції мови C++: перевантаження та наслідування</i>	
	<i>Тема – 1. Використання перевантажених конструкцій</i>
6	Лекція 2.1. Перевантаження функцій Практичне заняття 2.1. Робота з перевантаженими функціями СРС 6. Обмеження при перевантаженні методів
7	Лекція 2.2. Перевантаження операторів Практичне заняття 2.2. Робота з перевантаженими операторами СРС 7. Обмеження при перевантаженні операторів
	<i>Тема – 2. Використання наслідування (успадкування) в класах</i>
8	Лекція 2.3. Наслідування: типи, можливості, обмеження Практичне заняття 3.1. Робота з наслідуваннями СРС 8. Наслідування від декількох базових класів. Порядок виклику конструкторів
9	СРС 9. Виникнення помилок у роботі деструкторів при множинному наслідуванні
<i>Розділ 4. Введення та виведення даних у C++. Робота з файлами</i>	
	<i>Тема – 1. Введення та виведення даних</i>
10	Лекція 3.1. Введення та виведення даних Практичне заняття 4.1. Введення та виведення даних: прапори форматів, маніпулятори вводу/виводу СРС 10. Експерименти з iostream з метою вивчення іо-можливостей
	<i>Тема – 2. Файлове введення та виведення даних</i>
11	Лекція 3.2. Розширені можливості вводу/виводу. Файлове введення та виведення даних Практичне заняття 4.2. Файлове введення та виведення даних СРС 11. Зберігання даних на зовнішніх носіях
12	СРС 12. Особливості роботи з бінарними файлами
<i>Розділ 5. Віртуальні функції, шаблони та виключення, RTTI</i>	
	<i>Тема – 1. Поліморфізм і віртуальні функції в програмах</i>
13	Лекція 4.1. Поліморфізм: поняття та використання. Віртуальні функції Лабораторна робота 1. Робота з віртуальними функціями Практичне заняття 5.1. Поліморфізм в програмах СРС 13. Віртуальний базовий клас, його необхідність

Термін виконання (тиждень)	Види робіт за розділами та темами
	<i>Тема – 2. Шаблони та обробка виключень</i>
14	<i>Лекція 4.2. Шаблони та обробка виключень: родові функції та класи, виключення в програмах</i> <i>Лабораторна робота 2. Робота з родовими функціями та класами</i> <i>Практичне заняття 5.2. Шаблони та обробка виключень</i> <i>СРС 14. Особливості використання шаблонів (templates)</i>
	<i>Тема – 3. Типи даних і дії з ними</i>
15	<i>Лекція 4.3. Ідентифікація типів даних (RTTI), оператори приведення типів</i> <i>Практичне заняття 5.3. Робота з операторами приведення типів</i> <i>СРС 15. Шаблон RTTI. Приклади застосування. Простори імен, функції перетворення типів</i>
<i>Розділ 6. Простори імен і шаблони проектування</i>	
	<i>Тема – 1. Простори імен і перетворення типів</i>
16	<i>Практичне заняття 6. Простори імен. Функції перетворення типів</i> <i>СРС 16. Багатомодульна програма. Особливості використання простору імен (namespace). Шаблони проектування</i>
	<i>Тема – 2. Шаблони проектування</i>
17	<i>СРС 17. Бібліотека STL. Базові можливості</i>
<i>Розділ 7. Додаткові відомості мови C++. Структури даних, хеш-таблиці</i>	
18	<i>СРС 18. Генерація тексту з заданими характеристиками за допомогою Марківського алгоритму. Інші можливості генерації тексту</i> <i>Проведення модульної контрольної роботи</i>

6. Самостійна робота студента

Самостійна робота студента в обох кредитних модулях дисципліни «Основи програмування» поділена на завдання відповідно до розділів і тем кожного з модулів. Самостійна робота складається з завдань, які студенти мають виконати самостійно в поза аудиторні години навчання. Завдання містять як теоретичні, так і практичні задачі та розташовані в календарному плані дисципліни так, щоб доповнювати лекційний матеріал або передувати виконанню практичних занять та лабораторних робіт.

Далі наведений перелік завдань за термінами їх виконання протягом семестрів.

6.1. Календарний план запланованої самостійної роботи на 1 семестр - «Основи програмування – 1. Базові конструкції»

Термін виконання (тиждень)	Види робіт за розділами та темами
1	<i>СРС 1. Встановлення компілятора GCC в середовищі Cygwin і MinGW</i>
2	<i>СРС 2. Можливості використання уявних типів даних</i>
3	<i>СРС 3. Випадки використання циклів з перед- та післяумовами (do і do-while)</i>
4	<i>СРС 4. Оператори управління: розгалуження та переходи. Робота з вкладеними циклами. Оптимізація вкладених циклів з метою підвищення ефективності коду</i>
5	<i>СРС 5. Оператори управління: цикли. Продовження. Робота з вкладеними циклами. Продовження. Використання операторів break і continue при роботі з циклами</i>
6	<i>СРС 6. Різниця між getchar(), gets() і scanf(), можливості їх комбінування з printf()</i>
7	<i>СРС 7. Робота функції return: основні поняття, виключення та помилки</i>
8	<i>СРС 8. Функції. Частина друга: створення та використання власних функцій.</i>

Термін виконання (тиждень)	Випадки, в яких використання рекурсії є обов'язковим
Термін виконання (тиждень)	Види робіт за розділами та темами
9	СРС 9. Робота з даними типу enum
10	СРС 10. Одно-, дво- та багатовимірні масиви: основні відомості
11	СРС 11. Вказівники. Частина друга: використання вказівників для роботи з масивами. Аналіз проблем і повідомлень компілятора, що виникають при використанні неініційованих вказівників
12	СРС 12. Вимірювання часу, який витрачається програмою при використанні різних типів сортувань
13	СРС 13. Які бувають помилки та причини їх виникнення при використанні складеного літералу
14	СРС 14. Використання UTF-8 зокрема символів кириличної абетки у програмі
15	СРС 15. Ознаки «хорошого» інтерфейсу
16	СРС 16. Порівняння двійкового та текстового режимів
17	СРС 17. Використання конструкцій “вказівник на вказівник”. Коректне розіменування вказівників
18	СРС 18. Масиви структур: використання та застосування

6.1. Календарний план запланованої самостійної роботи на 2 семестр - «Основи програмування – 2. Модульне програмування»

Термін виконання (тиждень)	Види робіт за розділами та темами
1	СРС 1. Прості абстрактні структури даних. Структури даних на прикладах: масиви та списки, робота з деревами, хеш-структури
2	СРС 2. Випадки «тупикових» ситуацій при роботі з деревами
3	СРС 3. Особливості використання оператора “<<” з різними типами даних
4	СРС 4. Відмінність у використанні struct і class
5	СРС 5. Безпечні масиви. Обробка ситуацій виходу за межі масиву
6	СРС 6. Обмеження при перевантаженні методів
7	СРС 7. Обмеження при перевантаженні операторів
8	СРС 8. Наслідування від декількох базових класів. Порядок виклику конструкторів
9	СРС 9. Виникнення помилок у роботі деструкторів при множинному наслідуванні
10	СРС 10. Експерименти з iostream з метою вивчення іо-можливостей
11	СРС 11. Зберігання даних на зовнішніх носіях
12	СРС 12. Особливості роботи з бінарними файлами
13	СРС 13. Віртуальний базовий клас, його необхідність
14	СРС 14. Особливості використання шаблонів (templates)
15	СРС 15. Шаблон RTTI. Приклади застосування. Простори імен, функції перетворення типів
16	СРС 16. Багатомодульна програма. Особливості використання простору імен (namespace). Шабини проектування
17	СРС 17. Бібліотека STL. Базові можливості
18	СРС 18. Генерація тексту з заданими характеристиками за допомогою Марківського алгоритму. Інші можливості генерації тексту

4. Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

У навчальній дисципліні «Основи програмування» протягом семестрів передбачено проведення: лекційних і практичних занять (комп'ютерних практикумів) та лабораторних робіт.

Відвідування занять

Відвідування усіх видів аудиторних занять, а саме: лекційних, практичних (комп'ютерних практикумів) і лабораторних є обов'язковим.

Виняток становить лише відсутність на заняттях з поважних причин (тобто хвороби, форм-мажорні обставини, участь в олімпіадах, тощо), про що студент має повідомити викладача та надати підтвердження причини. У разі відсутності студента на аудиторному занятті з поважних причин нарахування штрафних балів не здійснюється.

Правила захисту лабораторних і практичних робіт

Протягом семестру студенти мають виконувати і захищати лабораторні та практичні роботи. Для кожної з робіт встановлені граничні терміни захисту (дедлайни).

Для отримання максимальних балів за кожну з передбачених програмою робіт студент повинен виконати роботу в повній мірі, без помилок і вчасно її захистити.

Якщо студент пропустив термін захисту роботи з поважної причини, він має можливість виконати та здати роботу без втрати балів.

Якщо студент пропустив термін захисту роботи без поважних причин – йому надається можливість відпрацювання пропущеної роботи шляхом написання самостійної роботи за відповідною темою. Але в такому випадку максимальний бал за роботу становитиме 75% від найвищої оцінки за завдання.

Причиною, що унеможлиблює захист пропущених робіт є лише закінчення навчального процесу (семестру).

Політика дедлайнів та перескладань

На кожну лабораторну та практичну роботу за календарним планом кожного з кредитних модулів дисципліни виділяється по два тижні для виконання та захисту роботи. Всі терміни оголошуються студентам на початку кожного семестру.

Перескладання лабораторних і практичних робіт не передбачено. В кінці кожного семестру передбачено перескладання модульної контрольної роботи.

8. Види контролю та рейтингова система оцінювання результатів навчання

Для кожного з кредитних модулів дисципліни передбачені наступні види контролю:

- проведення поточного контролю;
- проведення календарного контролю;
- проведення семестрового контролю.

Поточний контроль

Протягом семестру проводиться поточний контроль у вигляді експрес-опитування та модульної контрольної роботи. Поточний контроль має за мету перевірку знань та вмінь студентів, засвоєних та здобутих ними при прослуховуванні лекційних занять, виконанні практичних, самостійних та лабораторних робіт.

Експрес-опитування проводяться за матеріалами лекційних занять і складаються з коротких теоретичних запитань.

Модульні контрольні роботи проводяться за матеріалами усіх аудиторних видів занять, тобто: лекційних і практичних занять, лабораторних робіт. Завдання складаються з теоретичних і практичних запитань. Контрольні роботи проводяться в кінці кожного семестру. Питання та опорні теми, а також розділи дисципліни для повторення оголошуються студентам заздалегідь.

Для проведення модульної контрольної роботи виділяються дві учбові години за рахунок лабораторного заняття в першому семестрі та дві учбові години за рахунок лекційного заняття у другому семестрі.

Самостійні роботи проводяться за матеріалами лекційних занять і лабораторних робіт, складаються з теоретичних і практичних завдань, проводяться за розділами дисципліни. Самостійні роботи передбачені для проведення календарного контролю та для відпрацювання пропущених без поважних причин лабораторних занять.

Календарний контроль

Календарний контроль проводиться двічі на семестр як моніторинг поточного стану знань та вмінь студентів і виконання ними умов силабусу (навчальної програми) дисципліни.

Умовою отримання атестації (позитивної оцінки) за проходження календарного контролю є виконання вимог за кількістю та якістю виконаних лабораторних робіт, а також отримання позитивної оцінки за написання відповідної самостійної роботи.

Для отримання позитивної оцінки першої атестації студент повинен набрати мінімально 15 балів, для другої атестації 28 балів.

Причинами, що призводять до отримання незадовільного результату атестації під час календарного контролю є отримання незадовільної оцінки за самостійну роботу та захист передбачених календарним планом лабораторних робіт на незадовільні бали або ж у кількості менше 60% запланованих.

Семестровий контроль

Семестровий контроль з дисципліни «Основи програмування» в обох семестрах здійснюється у вигляді екзамену.

Умовами допуску до здачі екзамену є виконання усіх завдань, передбачених лабораторними та практичними роботами. А також, написання самостійних робіт календарного контролю та модульної контрольної роботи на оцінки не нижче оцінки «достатньо» (60% від максимального балу). Загальний рейтинг студента за семестр для допуску до екзамену має становити не менше 36 балів.

Екзамени проводяться у письмовому вигляді, містять теоретичні питання та практичні завдання. Питання, які виносяться на екзамен заздалегідь надаються студентам.

Система оцінювання результатів навчання

Оцінювання результатів навчання з дисципліни «Основи програмування» здійснюється згідно рейтингової системи оцінювання за 100-бальною шкалою.

Рейтинг студента складається з балів, які він отримав за виконання та захист лабораторних і практичних робіт, написання самостійних робіт і модульної контрольної роботи та складання іспиту.

Також, під час роботи на лекційних і практичних заняттях і під час захисту лабораторних робіт кожен студент може отримати заохочувальні додаткові бали. За невиконання умов навчання передбачені штрафні бали. Кількість заохочувальних або штрафних балів рівна 5 балам на семестр.

Оцінювання лабораторних робіт

Для першого кредитного модуля дисципліни – «Основи програмування – 1. Базові конструкції» - передбачено проведення 6 лабораторних робіт. Для другого кредитного модуля дисципліни – «Основи програмування – 2. Модульне програмування» – передбачено проведення 2 лабораторних робіт і 6 практичних.

У першому кредитному модулі лабораторні роботи оцінюються в 8 балів кожна за наступними критеріями:

– бездоганна робота оцінюється в 8 балів – завдання виконано в повній мірі, вирішує всі поставлені задачі, виконано без помилок або є лише незначні недоліки;

– є певні недоліки в виконанні роботи – 7-1 бал – завдання виконано неповністю, допущені суттєві помилки;

– робота не виконана, не захищена або не зданий звіт – 0 балів – робота здана лише у вигляді звіту, але не захищена особисто; робота здана особисто, але не має звіту; робота виконана менш, ніж на 60% або здана пізніше дедлайну.

У другому кредитному модулі лабораторні роботи оцінюються в 3 бали кожна за наступними критеріями:

– бездоганна робота оцінюється в 3 бали – завдання виконано в повній мірі, вирішує всі поставлені задачі, виконано без помилок або є лише незначні недоліки;

– є певні недоліки в виконанні роботи – 2-1 бал – завдання виконано неповністю, допущені суттєві помилки;

– робота не виконана, не захищена або не зданий звіт – 0 балів – робота здана лише у вигляді звіту, але не захищена особисто; робота здана особисто, але не має звіту; робота виконана менш, ніж на 60% або здана пізніше дедлайну.

Практичні роботи в другому кредитному модулі оцінюються в 7 балів за критеріями аналогічними до лабораторних робіт першого кредитного модуля за таким розподілом:

– бездоганна робота оцінюється в 7 балів – завдання виконано в повній мірі, вирішує всі поставлені задачі, виконано без помилок або є лише незначні недоліки;

– є певні недоліки в виконанні роботи – 6-1 бал – завдання виконано неповністю, допущені суттєві помилки;

– робота не виконана, не захищена або не зданий звіт – 0 балів – робота здана лише у вигляді звіту, але не захищена особисто; робота здана особисто, але не має звіту; робота виконана менш, ніж на 60% або здана пізніше дедлайну.

Оцінювання самостійних робіт

Самостійні роботи призначені для проведення календарного контролю оцінюються в 5 балів. Оцінювання самостійних робіт відбувається за такими ж критеріями, що й оцінювання модульної контрольної роботи (описані далі).

Самостійні роботи, що виконуються студентами з умовою відпрацювання пропущених робіт без поважної причини оцінюються за критеріями відповідними лабораторним і практичним роботам, без здачі звіту.

Оцінювання модульної контрольної роботи

Контрольна робота складається із чотирьох завдань: двох теоретичних питань і двох практичних завдань. Оцінювання роботи виконується за однаковими критеріями, незалежно від семестру, та з різними балами. Теоретичні питання оцінюються в 2 бали кожне, практичні по 4 бали кожне.

Оцінюються питання модульної контрольної роботи за такими критеріями:

– «відмінно», повна відповідь, не менше 90% потрібної інформації, програмний код написаний згідно стандартів кодування, вирішує основне завдання, включає всі необхідні програмні модулі, конструкції та хоча б частину перевірок; відповіді написані точними формулюваннями та розгорнуто;

– «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами, є незначні неточності, відсутні перевірки та код неструктурований; відповіді короткі, але точні;

– «задовільно», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з певними недоліками, в коді відсутні перевірки, наявні помилки; відповіді неповні;

– «незадовільно», відповідь не відповідає умовам до «достатньо», наявні суттєві помилки, відповідь відсутня або не вирішує поставлені задачі.

Оцінювання екзаменаційної роботи

Умовою допуску до екзамену є зарахування всіх лабораторних робіт, написання модульної контрольної роботи і стартовий рейтинг не менше 36 балів.

Екзамен студенти виконують письмово. Білети складаються з теоретичних і практичних завдань. Теоретичні запитання оцінюються по 8 балів кожне, практичні – у 12 балів кожне, за такими критеріями:

- «відмінно», повна відповідь, не менше 90% потрібної інформації, що виконана згідно з вимогами до рівня вмінь;
- «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами до рівня вмінь або є незначні неточності;
- «задовільно», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з певними недоліками;
- «незадовільно», відповідь не відповідає умовам до «задовільно» – 0 балів.

Сума стартових балів та балів за екзаменаційну роботу переводиться до екзаменаційної оцінки згідно з таблицею:

Бали	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

Робочу програму навчальної дисципліни (силабус):

Складено:

доцент, к.т.н., Шпурик Вадим Вадимович,
асистент кафедри АПЕПС Оленева Ксенія Миколаївна

Ухвалено кафедрою АПЕПС (протокол № 16 від 18.06.2021 р.)

Погоджено Методичною комісією ТЕФ КПІ ім. Ігоря Сікорського ¹ (протокол № 11 від 24.06.2021 р.)