



Об'єктно-орієнтоване програмування.

Частина 1. Синтаксис мов програмування С/С++

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>
Освітня програма	<i>"СИСТЕМНЕ ПРОГРАМУВАННЯ ТА СПЕЦІАЛІЗОВАНІ КОМП'ЮТЕРНІ СИСТЕМИ"</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>2 курс, осінній семестр</i>
Обсяг дисципліни	<i>105 год / 3,5 кредитів ЕКТС (лекції – 36 год, лабораторні роботи – 18 год, СРС – 51 год)</i>
Семестровий контроль/ контрольні заходи	<i>Залік, МКР, ДКР</i>
Розклад занять	https://schedule.kpi.ua/
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: к.т.н., асистент каф. СПСКС, Морозов Костянтин В'ячеславович, morozov.kostiantyn@ill.kpi.ua Лабораторні: к.т.н., асистент каф. СПСКС, Морозов Костянтин В'ячеславович, morozov.kostiantyn@ill.kpi.ua
Розміщення курсу	Посилання на дистанційний ресурс (Google Classroom: https://classroom.google.com/c/NjgwNzk3MDI4NzY1?cjc=fezbbsp)

Програма навчальної дисципліни

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

Об'єкто-орієнтоване програмування (ООП) – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. С++ – мова програмування загального призначення з підтримкою різних парадигм програмування: об'єктно-орієнтованої, узагальненої, процедурної тощо, яка бере витоки з мови програмування С. С++ є однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки прикладного програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм. Вивчення кредитного модуля «Об'єктно-орієнтоване програмування 1. Синтаксис мов програмування С/С++» дозволяє сформувати у студентів компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної з виконанням проектування інтерфейсів й кодування функцій і класів, створенням й налагодженням програмних проектів.

Метою кредитного модуля є формування у студентів здатностей:

- розробки програм мовами С та С++;
- вміння проектування інтерфейсу й кодування функцій;
- вміння розміщення об'єктів у динамічній пам'яті;
- створення й налагодження програмних проєктів;
- проєктування програм згідно з принципами ООП.

Предметом кредитного модуля є синтаксис, основні елементи, способи та прийоми програмування мовами С та С++, а також принципи ООП.

Дисципліна сприяє формуванню у студентів таких компетентностей:

- ЗК1 Здатність до абстрактного мислення, аналізу та синтезу
- ЗК3 Здатність застосовувати знання у практичних ситуаціях
- ЗК7 Вміння виявляти, ставити та вирішувати проблеми
- ФК2 Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення
- ФК3 Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж
- ФК15 Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення
- ФК16 Здатність до алгоритмічного та логічного мислення

Формування зазначених компетентностей забезпечує досягнення наступних програмних результатів навчання:

- ПРН3 Знати новітні технології в галузі комп'ютерної інженерії
- ПРН6 Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.
- ПРН7 Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.
- ПРН9 Вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності.
- ПРН10 Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

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

- Дисципліні «Об'єктно-орієнтоване програмування. Частина 1» (ПО 7.1) передують дисципліни «Програмування. Частина 1.» (ЗО 9.1),

“Програмування. Частина 2.” (ЗО 9.2), “Структури даних та алгоритми. Частина 1.” (ПО 2.1) та “Структури даних та алгоритми. Частина 2.” (ПО 2.2).

- Кредитний модуль «Об’єктно-орієнтоване програмування. Частина 1» (ПО 7.1) забезпечує вивчення забезпечує вивчення другої частини цього курсу «Об’єктно-орієнтоване програмування. Частина 2» (ПО 7.2), кредитного модуля “Паралельне програмування” (ПО8).

3. Зміст навчальної дисципліни

- Тема 1. Основи мов С та С++.
- Тема 2. Класи та принципи ООП.
- Тема 3. Інші можливості мови С++.
- Тема 4. Шаблони в С++.
- Тема 5. Розширені теми.

4. Навчальні матеріали та ресурси

Базова література

1. Трофименко О.Г. С++. Основи програмування. Теорія та практика [Текст] / О.Г. Трофименко – Одеса: «Фенікс», 2010 – 544 с.
2. Васильєв О. Програмування на С++ в прикладах і задачах [Текст] / О. Васильєв – «Ліра-К», 2017 – 382 с.
3. Шпак З. Програмування мовою С [Текст] / З. Шпак – Львів: «Львівська політехніка», 2011 – 436 с.
4. Бублик В.В. Об’єктно-орієнтоване програмування [Текст] / В.В. Бублик – К.: «ІТ-книга», 2015 – 624 с.: іл.
5. Львов М.С., Співаковський О.В. Вступ до об’єктно-орієнтованого програмування. Навчальний посібник [Текст] / М.С. Львов, О.В. Співаковський – Херсон: ХГПУ, 2000 – 238 с.: іл.

Допоміжна література

6. Brian W. Kernighan, Dennis M. Ritchie. C Programming Language, 2nd Edition / Pearson; 2nd edition (March 22, 1988) – 272 p.
7. Bjarne Stroustrup. Programming: Principles and Practice Using C++, 2nd Edition / Addison-Wesley Professional, 2014 – 1312 p.
8. Stephen Prata. C++ Primer Plus (Developer's Library) 6th Edition / Addison-Wesley Professional; 6th edition (October 18, 2011) – 1440 p.
9. Nicolai M. Josuttis. The C++ Standard Library - A Tutorial and Reference, 2nd Edition / Addison Wesley Longman, 2012 – 1136 p.
10. Matt Weisfeld. Object-Oriented Thought Process, The (Developer's Library) 4th Edition / Addison-Wesley Professional; 4th edition (March 13, 2013) – 336 p.
11. David Vandevoorde, Nicolai M. Josuttis, Douglas Gregor. C++ Templates: The Complete Guide 2nd Edition / Addison-Wesley Professional; 2nd edition (September 14, 2017) – 822 p.
12. Radovan Bast, Roberto Di Remigio. CMake Cookbook: Building, testing, and packaging modular software with modern CMake / Packt Publishing (September 26, 2018) – 606 p.

Інформаційні ресурси

1. Google-диск. – Режим доступу:

https://drive.google.com/drive/folders/11cCNc8SwjeoAOYvp361OvRng5rwc2aZi?usp=drive_link

2. Електронний кампус НТУУ «КПІ». Матеріали з дисципліни «Об'єктно-орієнтоване програмування». – Режим доступу : <https://campus.kpi.ua/>

3. C and C++ reference: <https://en.cppreference.com/w/>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна охоплює 36 годин лекцій та 18 годин лабораторних робіт, а також виконання модульної контрольної роботи та домашньої контрольної роботи.

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань
Тема 1. Основи мов С та С++	
1	Мова С: основи. Типи, змінні, структури, перечислення, арифметико-логічні операції.
2	Мова С: можливості для написання складних програм. Умовні оператори, цикли, функції. Директиви препроцесора та умовна компіляція. Аргументи командного рядка. Заголовні файли.
3	Огляд можливостей мови С++ . Логічний тип, посилання, простори імен. Перевантаження функцій. Анонімні (лямбда) функції. Перетворення типів. Робота з динамічною пам'яттю.
Тема 2. Класи та принципи ООП	
4	Класи та об'єкти. Поняття класу. Класи та об'єкти. Конструктори й деструктор, методи та поля класу. Статичні члени класу. Специфікатори доступу. Константні методи.
5	Класи в мові С++. Перевантаження операторів. Дружні функції та класи. Користувачькі літерали. Функціональні об'єкти та лямбда-функції.
6	Успадкування класів. Базові класи та класи-нащадки. Реалізація успадкування в мові С++. Послідовність ініціалізації полів. Доступ до членів класів у ієрархії.
7	Поліморфізм. Поліморфні класи. Віртуальні методи та віртуальний деструктор. Абстрактні класи та абстрактні (чисто віртуальні) функції. Реалізація поліморфізму в С++. Чисто віртуальні функції та абстрактні класи. Динамічне перетворення типів.
8	Принципи ООП/ООД. Принципи ООП та ООД. Поняття, пов'язані з ООП. Принципи SOLID та патерни GRASP.
9	Складні випадки успадкування. Динамічне перетворення типів. Множинне успадкування. Ромбовидне успадкування. Віртуальне успадкування. Реалізація множинного та віртуального успадкування.
Тема 3. Інші можливості мови С++	
10	Обробка виняткових ситуацій. Винятки. Генерація та перехоплення винятків. Обробка винятків в мові С++. Проблема контролю над ресурсами та ідіома RAII.

11	Семантика переміщення. Копіювання та переміщення. Реалізація семантики переміщення в мові C++. Константність часу компіляції та часу виконання.
Тема 4. Шаблони та узагальнення	
12	Шаблони та узагальнення. Призначення шаблонів. Реалізація шаблонів в C++. Шаблони класів та функцій. Параметри шаблонів.
13	Спеціалізація шаблонів. Повна та часткова спеціалізація шаблонів. CRTP.
14	Поліморфізм на основі шаблонів. Статичний та динамічний поліморфізм. Інстанціювання шаблонів.
15	Складні випадки використання шаблонів. Залежні типи в шаблонах. Шаблони як параметри шаблонів. Варіадичні шаблони. SFINAE. Основи метапрограмування в мові C++.
Тема 5. Розширені теми	
16	Інші можливості мови C++. Згортання rvalue-посилань та автоматичне виведення типів. Універсальна ініціалізація.
17	Утиліта CMake: основи. Синтаксис та призначення CMake. Змінні, умовні оператори, цикли, макроси, функції.
18	Утиліта CMake: налаштування збірки. Цілі. Підпроекти. Бібліотеки. Визначення препроцесора то опції компіляції.

Лабораторні роботи (комп'ютерні практикуми)

Основне завдання циклу лабораторних робіт циклу лабораторних є набуття студентами уміння ефективного кодування програмних додатків та практичного досвіду з побудови й налагодження проектів.

№ з/п	Назва та завдання лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	Розробка простого класу. Завдання: Спроекувати та реалізувати клас згідно із варіантом. Результати роботи оформити у вигляді проекту C++-програми та звіту.	6
2	Розробка сімейства поліморфних класів. Завдання: Спроекувати та реалізувати ієрархію класів згідно із варіантом. Результати роботи оформити у вигляді проекту C++-програми та звіту.	4
3	Розробка шаблонів функцій та класів. Завдання: Спроекувати та реалізувати шаблони класів та функцій згідно із варіантом. Результати роботи оформити у вигляді проекту C++-програми та звіту.	4
4	Розробка класу колекції. Завдання: Спроекувати та реалізувати клас колекції згідно із варіантом. Передбачити обробку виключних ситуацій. Результати роботи оформити у вигляді проекту C++-програми та звіту.	4

6. Самостійна робота

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1	Проекти Microsoft Visual Studio Типи проектів MSVC++. Створення порожнього проекту. Додавання файлів, Компіляція проекту.	2
2	Налаштування параметрів проектів в Microsoft Visual Studio Вибір версії компілятора (версія стандарту мови C++). Параметри оптимізації. Рівень попереджень. Передача параметрів командного рядка. Визначення прероцесора.	2
3	Налаштування редактора в Microsoft Visual Studio Встановлення параметрів форматування та відображення. Автоматичне форматування коду.	2
4	Попередньо скомпільовані заголовні файли Призначення попередньо скомпільованих заголовних файлів. Налаштування та використання попередньо скомпільованих заголовних файлів.	2
5	Конфігурації рішення в Microsoft Visual Studio Налаштування конфігурацій для налагодження (Debug) та випуску (Release). Конфігурація для випуску із інформацією для налагодження (Release with debug info). Налаштування платформи (Win32, x64, ARM).	2
6	Засоби відналагодження та профілювання в Microsoft Visual Studio Відналагодження програми в Microsoft Visual Studio. Точки зупинки, умовні точки зупинки. Стек викликів. Засоби профілювання та їх використання зокрема для оптимізації швидкодії роботи програми та моніторингу використання пам'яті.	3
7	Арифметичні перетворення типів. Перевантаження арифметичних операторів. Перетворення типів при виконанні алгебраїчних операцій.	2
8	Директиви препроцесора та умовна компіляція Директива препроцесора #pragma. Пригнічення повідомлень про деякі групи попереджень (warnings). Вирівнювання полів у структурах (#pragma pack).	2
9	Кодування та налагодження лямбда-функцій Реалізація лямбда функцій. Список захоплення, різні способи захоплення змінних з контексту.	2
10	Кодування та налагодження простих класів. Реалізація простих класів: поля, методи, конструктори, деструктор, перевантаження операторів. Контроль доступу до членів класу.	2
11	Тестування можливостей успадкування класів. Реалізація успадкування класів: приватне, захищене, публічне. Доступ до членів класів при успадкуванні.	2
12	Кодування та налагодження ієрархії класів. Реалізація віртуальних методів. Реалізація абстрактного базового класу.	2
13	Дослідження порядку виклику конструкторів та деструкторів.	2

	Конструювання та знищення полів класу. Виклик конструкторів та деструкторів при успадкуванні.	
14	Тестування механізму динамічного перетворення типів. Динамічне перетворення типів для посилань та для вказівників. Випадки успішного та неуспішного перетворення типів.	2
15	Тестування можливостей перехоплення виключень. Генерація та перехоплення виключень. Перехоплення виключень різних типів.	2
16	Віртуальні та невіртуальні методи Порівняння особливостей роботи віртуальних та невіртуальних методів. Перевірка роботи ключових слів <code>override</code> та <code>final</code> .	2
17	Дослідження проблеми витоків пам'яті Витоки пам'яті у випадку неправильної обробки виключень. Витоки пам'яті за відсутності віртуального деструктора.	2
18	Реалізація «розумного» вказівника із застосуванням RAII Використання принципів RAII для реалізації вказівника із унікальним володінням (може бути лише один володар ресурсу) та автоматичним звільненням пам'яті.	2
19	Логічна та фізична константність Константні змінні, константні поля, константні методи. Використання ключового слова <code>mutable</code> . Константність методів та параметрів-посилань в інтерфейсах.	2
20	Кодування й налагодження шаблонів функцій. Параметри шаблону. Перевантаження та шаблони функцій.	2
21	Кодування й налагодження шаблонів класів. Спеціалізація шаблонів. Часткова спеціалізація.	2
22	Тестування можливостей інстанціювання шаблонів Інстанціювання шаблонів класів. Інстанціювання шаблонів функцій. Інстанціювання та спеціалізація.	2
23	Тестування можливостей метапрограмування Реалізація шаблонів для визначення характеристик класів ().	2
24	Генерація помилок етапу компіляції Конструкція <code>static_assert</code> та її використання.	2
25	Створення проекту за допомогою утиліти CMake Створення та конфігурація простого проекту на мові C++.	2

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- відвідування лекційних та лабораторних занять є обов'язковою складовою вивчення матеріалу;
- на лекції викладач користується власним презентаційним матеріалом; використовує гугл-класрум для викладання матеріалу поточної лекції, додаткових ресурсів, лабораторних робіт та інше;
- на лекції заборонено відволікати викладача від викладання матеріалу, усі питання, уточнення та ін. студенти задають в кінці лекції у відведений для цього час (окрім випадків, коли викладач сам пропонує задати питання під час лекції);

- лабораторні роботи захищаються у два етапи – перший етап: студенти виконують завдання на допуск до захисту лабораторної роботи; другий етап – захист лабораторної роботи. Бали за лабораторну роботу враховуються лише за наявності електронного звіту;
- заохочувальні бали виставляються за: активну участь у підготовці лекційних матеріалів; участь у факультетських та інститутських олімпіадах з навчальних дисциплін, участь у конкурсах робіт, підготовка оглядів наукових праць; презентацій по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 10;
- штрафні бали виставляються за: невчасне відпрацювання та здачу лабораторної роботи
- 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.
- 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 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.

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

Семестровий контроль проводиться у вигляді заліку. Для оцінювання результатів навчання застосовується 100-бальна рейтингова система та університетська шкала.

Поточний контроль: виконання та захист лабораторних робіт, написання модульної контрольної роботи, написання домашньої контрольної роботи.

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

Семестровий контроль: залік

Якщо семестровий рейтинг більше 60 балів студент може не виходити на залікову контрольну роботу, а отримати оцінку «автоматом».

Рейтинг студента з дисципліни складається з балів, що він отримує за:

1. виконання та захист 4 лабораторних робіт;
2. написання модульної контрольної роботи;
3. написання домашньої контрольної роботи;
4. заохочувальних та штрафних балів.

Лабораторні роботи

Загальна кількість балів – 60. Оцінка:

- «відмінно», повна відповідь на питання під час захисту (не менш ніж 90% потрібної інформації) та оформлений належним чином електронний протокол до лабораторної роботи – 14-15 балів;
- «добре», достатньо повна відповідь на питання під час захисту (не менш ніж 75% потрібної інформації) та оформлений належним чином електронний протокол до лабораторної роботи – 11-13 балів;
- «задовільно», неповна відповідь на питання під час захисту (не менш ніж 60% потрібної інформації), незначні помилки та оформлений належним чином електронний протокол до лабораторної роботи – 9-10 балів;
- «достатньо», неповна відповідь на питання під час захисту (не менш ніж 30% потрібної інформації), незначні помилки та оформлений належним чином електронний протокол до лабораторної роботи – 5-8 балів;
- «незадовільно», незадовільна відповідь та/або не оформлений належним чином електронний протокол до лабораторної роботи – 0-4 бали.

За кожне запізнення з поданням лабораторної роботи до захисту від встановленого терміну оцінка знижується на 1 бал.

Заохочувальні бали

– за виконання творчих робіт з кредитного модуля (наприклад, участь у факультетських та інститутських олімпіадах з навчальних дисциплін, участь у конкурсах робіт, підготовка оглядів наукових праць тощо); за активну роботу на лекції (питання, доповнення, зауваження за темою лекції, коли лектор пропонує студентам задати свої питання); виконання додаткових завдань у лабораторних роботах 1-3 бали, але в сумі не більше 10;

– презентації по СРС – від 1 до 5 балів.

Домашня контрольна робота

Загальна кількість балів – 10. Домашня контрольна робота містить одне практичне завдання: побудова ієрархії класів згідно із індивідуальним завданням. Обов'язковим є використання ієрархій поліморфних класів. Особливу увагу необхідно приділити розробці інтерфейсів, зокрема, константності параметрів та методів. Клас-колекцію для збереження елементів реалізувати самостійно у вигляді шаблону класу.

Приклади варіантів індивідуальних завдань для ДКР запропоновані в Додатку А. Завдання також можуть бути запропоновані здобувачем освіти самостійно та обов'язково затверджені викладачем.

Оцінка:

«відмінно», повна відповідь (не менш ніж 90% потрібної інформації) – 9-10 балів;

«добре», достатньо повна відповідь (не менш ніж 75% потрібної інформації), або повна відповідь з незначними помилками – 8 балів;

«задовільно», неповна відповідь (але не менш ніж 60% потрібної інформації) та незначні помилки – 6-7 балів;

«достатньо», неповна відповідь (не менш ніж 30% потрібної інформації), незначні помилки – 3-6 балів;

«незадовільно» - 0-2 бали. Незадовільна відповідь (неправильний розв'язок задачі).

Модульна контрольна робота

Загальна кількість балів – 30. Модульна контрольна робота містить два теоретичних запитання та одне практичне завдання. Кожне запитання (завдання) оцінюється у 10 балів. Перелік теоретичних питань наведений у додатках.

Оцінка:

«відмінно», повна відповідь (не менш ніж 90% потрібної інформації) – 27-30 балів;

«добре», достатньо повна відповідь (не менш ніж 75% потрібної інформації), або повна відповідь з незначними помилками – 23-25 балів;

«задовільно», неповна відповідь (але не менш ніж 60% потрібної інформації) та незначні помилки – 18-22 балів;

«достатньо», неповна відповідь (не менш ніж 30% потрібної інформації), незначні помилки – 9-17 балів;

«незадовільно» - 0-8 балів. Незадовільна відповідь (неправильний розв'язок задачі), потребує обов'язкового повторного написання в кінці семестру.

Календарний контроль

За результатами навчальної роботи за перші 7 тижнів максимально можлива кількість балів – 15 (1 лабораторна робота). На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менший ніж 7 балів.

За результатами 14 тижнів навчання максимально можлива кількість балів – 75 балів (3 лабораторні + МКР. На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менший ніж 37 балів.

Максимальна сума вагових балів контрольних заходів протягом семестру складає:

$$RD = 4 \cdot r_{\text{лаб}} + r_{\text{МКР}} + r_{\text{ДКР}} + (r_z - r_{\text{ш}}) = 100 + (r_z - r_{\text{ш}}),$$

де $r_{\text{лаб}}$ – бал за лабораторну роботу;

$r_{\text{МКР}}$ – бал за написання МКР;

$r_{\text{ДКР}}$ – бал за написання ДКР;

r_z – заохочувальні бали за активну участь на лекціях, презентації, участь в олімпіадах, конкурсі роботи, наукові роботи за тематикою дисципліни (0...10);

$r_{\text{ш}}$ – штрафні бали.

Залік

Умовою допуску до заліку є зарахування всіх лабораторних робіт, та стартовий рейтинг не менше 36 балів.

На заліку студенти виконують письмову контрольну роботу. Кожен білет містить два теоретичних запитання та одне практичне завдання (задачу). Перелік тем для теоретичних питань наведений у додатку Б. Кожне теоретичне запитання оцінюється у 30 балів. Практичне завдання оцінюється в 40 балів.

Система оцінювання питань

Загальна кількість балів – 100. Оцінка:

«відмінно», повна відповідь, не менше 90% потрібної інформації, що виконана згідно з вимогами до рівня «умінь», (повне, безпомилкове розв'язування завдання) – 90-100 балів;

«добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами до рівня «умінь» або є незначні неточності (повне розв'язування завдання з незначними неточностями) – 75-89 балів;

«задовільно», неповна відповідь, не менше 60% потрібної інформації, що виконана згідно з вимогами до «стереотипного» рівня та деякі помилки (завдання виконане з певними недоліками) – 60-74 балів;

«достатньо», неповна відповідь на питання (не менш ніж 30% потрібної інформації) – 30-59 балів;

«незадовільно», відповідь не відповідає умовам до «задовільно» – 0-29 балів.

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

Семестровий контроль: залік

Умови допуску до семестрового контролю: мінімально позитивна оцінка за зарахування усіх лабораторних робіт семестровий рейтинг не менше 60 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)

Перелік питань, які виносяться на семестровий контроль представлено у Додатку Б.

З метою кращого засвоєння окремих тем пропонується проходження відповідних онлайн-курсів, зокрема:

<https://www.coursera.org/learn/codio-cpp-basics>

<https://www.coursera.org/learn/cpp-object-basics>

<https://www.coursera.org/learn/object-oriented-cpp>

<https://www.coursera.org/learn/c-interacting-with-the-world-and-error-handling>

<https://www.coursera.org/learn/c-plus-plus-a>

а також інших, що відповідають тематиці курсу.

Робочу програму навчальної дисципліни (силабус):

Складено асистент каф. СПКС, к.т.н, Морозов Костянтин В'ячеславович

Ухвалено кафедрою __СПКС__ (протокол № _11_ від _12.06.2024_)

Погоджено Методичною комісією факультету (протокол № _12_ від 21.06.2024_)

Додаток А

Приклади індивідуальних завдань для домашньої контрольної роботи

- Потяг (номер потягу; тип потягу: вантажний/пасажирський/змішаний; кількість вагонів, вивід інформації про всі вагони), локомотив (тип: дизель/електричний, потужність), вантажний вагон (вантажопідйомність, маса вантажу, тип вантажу), пасажирський вагон (кількість місць, кількість пасажирів).
- Депо (кількість транспортних засобів; тип: автобусне, трамвайне, тролейбусне, змішане; вивід інформації про всі транспортні засоби), трамвай (кількість пасажирів; кількість вагонів; модель; потужність; бортовий номер), тролейбус (кількість пасажирів; потужність; бортовий номер), автобус (кількість пасажирів; потужність; тип палива: бензин, дизель, газ; номер).
- Вулиця (кількість будинків; кількість жителів; вивести інформації про всі будинки), багатоквартирний будинок (кількість квартир; кількість поверхів; кількість жителів; номер будинку), приватний будинок (кількість жителів; кількість поверхів; житлова площа; номер будинку), нежитлова будова (кількість поверхів; тип: завод, магазин, школа; номер будинку).

4. Кошик товарів (кількість товарів; загальна вартість; тип товарів: харчові/нехарчові/змішані; вивід інформації про всі товари), вагові товари (вага; тип: борошно/сіль/цукор/круп/солодощі/спеції; ціна за кілограм; ціна), напої (об'єм; тип: мінеральна вода, солодка вода, пиво, молоко; ціна), нехарчові товари (тип: мило, порошок, зубна паста; ціна).
5. Склад комп'ютерних комплектуючих (кількість; загальна вартість; тип комплектуючих; вивід інформації про всі комплектуючі), процесор (частота; обсяг кешу; вартість; виробник: Intel/AMD), оперативна пам'ять (обсяг пам'яті; частота; тип; вартість), відеокарта (обсяг пам'яті; тип пам'яті; вартість).

Додаток Б

Перелік теоретичних питань на залік

1. Оголошення та ініціалізація змінних.
2. Типи даних, складні типи даних, вказівники, масиви, логічний тип, nullptr_t.
3. Модифікатори const та volatile (cv-кваліфікатори).
4. Перечислення (в т.ч. типізовані перечислення).
5. Перевизначення типів (псевдоніми типів).
6. Перетворення типів, cast-конструкції.
7. Літерали. Користувачські літерали.
8. Рядки та символи. Неформатовані рядки.
9. Операції: арифметичні, логічні, порівняння, присвоєння.
10. Блоки коду, область видимості та існування локальних змінних.
11. Умовні оператори та цикли. Оператори виходу та продовження циклу. Тернарний оператор. Оператор вибору.
12. Функції, параметри, повернення значень. Перевантаження функцій. Параметри за замовченням. Альтернативний запис функцій.
13. Лямбда-функції. Захоплення контексту.
14. Локальні та глобальні змінні.
15. Директиви препроцесора. Макроси. Умовна компіляція.
16. Заголовні файли. Включення заголовних файлів. Особливості створення власних заголовних файлів.
17. Вказівники та посилання. Rvalue-посилання (семантика переміщення).
18. Робота з динамічною пам'яттю.
19. Простори імен. Анонімні простори імен та static.
20. Вивід типів: auto, decltype
21. Структури, об'єднання та класи.
22. Члени класів: поля та методи. Статичні члени.
23. Конструктори та деструктори.
24. Модифікатори доступу до членів класів.
25. Вкладені типи (в класах).
26. Неявний параметр this.
27. Константні методи.
28. Визначення членів поза межами визначення класу.
29. Дружні функції та дружні класи.
30. Перевантаження операторів. Оператори як методи класу та як окремі функції.
31. Оператори інкременту та декременту. Постфіксна та префіксна версії.
32. Успадкування класів. Типи успадкування.
33. Ініціалізація полів класу та базових класів. Послідовність ініціалізації (послідовність викликів конструкторів та деструкторів).
34. Перекриття імен. Використання using для вирішення проблеми перекриття імен.
35. Поліморфні класи. Віртуальні функції. Переваги та недоліки. Віртуальні деструктори. Заміщення (override). Віртуальні функції в конструкторах та деструкторах.
36. Реалізація механізму віртуальних функцій.
37. Чисто віртуальні функції, абстрактні класи, інтерфейси.

38. Заборона успадкування та заміщення.
39. Висхідне та низхідне перетворення типів, `static_cast` та `dynamic_cast` для роботи з ієрархіями класів.
40. Множинне успадкування та проблеми із ним пов'язані. Реалізація множинного успадкування.
41. Проблема ромбовидного успадкування. Віртуальне успадкування (віртуальний базовий клас) та його реалізація. Особливості віртуального успадкування (зокрема, конструювання).
42. Базові принципи ООП та суміжні поняття.
43. Принципи SOLID.
44. Патерни GRASP.
45. Механізм обробки виключень в C++. Генерація виключень та їх відловлення. Вплив виключень на ефективність роботи програми.
46. Ідіома RAII (Resource Acquisition Is Initialization).
47. Шаблони C++. Шаблони функцій, класів, псевдонімів. Параметри шаблонів.
48. Використання шаблонів. CRTP.
49. Повна та часткова спеціалізація шаблонів.
50. Інстанціювання шаблонів.
51. Складні випадки шаблонів. Залежні типи та використання шаблонів у шаблонах.
52. Варіадичні шаблони (зі змінною кількістю параметрів).
53. Метапрограмування. Принцип SFINAE.
54. Статичний та динамічний поліморфізм.