



Об'єктно-орієнтоване програмування.

Частина 2. Проектування складних об'єктів

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, весняний семестр</i>
Обсяг дисципліни	<i>123 год / 4 кредити ЕКТС (лекції – 36 год, лабораторні роботи – 18 год, СРС – 66 год)</i>
Семестровий контроль/ контрольні заходи	<i>Залік, МКР</i>
Розклад занять	https://schedule.kpi.ua/
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: к.т.н., асистент каф. СПКС, Морозов Костянтин В'ячеславович, morozov.kostiantyn@iit.kpi.ua Лабораторні: к.т.н., асистент каф. СПКС, Морозов Костянтин В'ячеславович, morozov.kostiantyn@iit.kpi.ua
Розміщення курсу	Посилання на дистанційний ресурс (Google Classroom: https://classroom.google.com/c/Njk2NDU2MzU1NjM4?cjc=6esr3tc)

Програма навчальної дисципліни

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

В даному кредитному модулі продовжується вивчення матеріалу, засвоєного в межах попереднього курсу «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ 1. СИНТАКСИС МОВ ПРОГРАМУВАННЯ С/С++». Окрему увагу присвячено патернам (або шаблонам) проектування, що описують типові способи вирішення поширених проблем при проектуванні програм. Основна частина курсу присвячена ознайомленню із можливостями стандартної бібліотеки мови С++ та отриманню практичного досвіду роботи з нею.

Вивчення кредитного модуля «Об'єктно-орієнтоване програмування 2. Проектування складних об'єктів» дозволяє сформувати у студентів компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної з виконанням декомпозиції програми на модулі для забезпечення ефективної розробки й супроводження програми.

Метою кредитного модуля є формування у студентів здатностей:

- аналізувати вимоги до програмної системи;
- обирати й ефективно застосовувати стандартні бібліотеки й класи;
- застосовувати засоби тестування й аналізу програм

Предметом кредитного модуля є принцип організації, основні елементи та прийоми використання стандартної бібліотеки мови C++ (STL), фреймворку Qt, а також патерни (шаблони) проєктування.

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 2» (ПО 7.2) передують дисципліни «Програмування. Частина 1.» (ЗО 9.1), «Програмування. Частина 2.» (ЗО 9.2), «Структури даних та алгоритми. Частина 1.» (ПО 2.1), «Структури даних та алгоритми. Частина 2.» (ПО 2.2) та

перша частина цього курсу «Об'єктно-орієнтоване програмування. Частина 1» (ПО 7.1).

- Кредитний модуль «Об'єктно-орієнтоване програмування. Частина 2» (ПО 7.2) забезпечує вивчення кредитного модуля «Паралельне програмування» (ПО8).

3. Зміст навчальної дисципліни

- Тема 1. Патерни проектування.
- Тема 2. Стандартна бібліотека шаблонів C++.
- Тема 3. Додаткові можливості стандартної бібліотеки C++.
- Тема 4. Використання сторонніх бібліотек.
- Тема 5. C++/CLI і .NET framework.

4. Навчальні матеріали та ресурси

Базова література

1. Фрімен Е., Робсон Е. Патерни проектування [Текст] / Е. Фрімен, Е. Робсон – Фабула, 2020 – 672 с.
2. Трофименко О.Г. C++. Основи програмування. Теорія та практика [Текст] / О.Г. Трофименко – Одеса: «Фенікс», 2010 – 544 с.
3. Васильєв О. Програмування на C++ в прикладах і задачах [Текст] / О. Васильєв – «Ліра-К», 2017 – 382 с.
4. Шпак З. Програмування мовою С [Текст] / З. Шпак – Львів: «Львівська політехніка», 2011 – 436 с.
5. Бублик В.В. Об'єктно-орієнтоване програмування [Текст] / В.В. Бублик – К.: «ІТ-книга», 2015 – 624 с.: іл.
6. 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

7. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language 3rd Edition / Addison-Wesley Professional; 3rd edition (September 15, 2003) – 208 p.
8. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch. Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition / Addison-Wesley Professional; 1st edition (November 10, 1994) – 416 p.
9. Brian W. Kernighan, Dennis M. Ritchie. C Programming Language, 2nd Edition / Pearson; 2nd edition (March 22, 1988) – 272 p.
10. Bjarne Stroustrup. Programming: Principles and Practice Using C++, 2nd Edition / Addison-Wesley Professional, 2014 – 1312 p.
11. Stephen Prata. C++ Primer Plus (Developer's Library) 6th Edition / Addison-Wesley Professional; 6th edition (October 18, 2011) – 1440 p.
12. Nicolai M. Josuttis. The C++ Standard Library - A Tutorial and Reference, 2nd Edition / Addison Wesley Longman, 2012 – 1136 p.
13. Matt Weisfeld. Object-Oriented Thought Process, The (Developer's Library) 4th Edition / Addison-Wesley Professional; 4th edition (March 13, 2013) – 336 p.
14. David Vandevoorde, Nicolai M. Josuttis, Douglas Gregor. C++ Templates: The Complete Guide 2nd Edition / Addison-Wesley Professional; 2nd edition (September 14, 2017) – 822 p.

15. 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/1TnNgwsfSp5adRl4eBcts3zqiLTpJdoVW?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/>
4. ПАТЕРНИ ПРОЕКТУВАННЯ – Режим доступу : <https://refactoring.guru/uk/design-patterns>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань
Тема 1. Патерни проектування.	
1	Патерни проектування, ч.1. Породжувальні патерни. Структурні патерни.
2	Патерни проектування, ч.2. Поведінкові патерни. Шаблон PImpl.
Тема 2. Стандартна бібліотека шаблонів C++.	
3	Вступ до STL. Пари та кортежі. Предикати та функціональні об'єкти. Допоміжні функції.
4	Послідовні контейнери. Масиви, вектори, списки, деки.
5	Впорядковані та неспорядковані контейнери. Множини та відображення. Мультимножини та мультिवідображення. Невпорядковані множини та неспорядковані відображення.
6	Спеціалізовані контейнери. Рядки. Адаптери контейнерів: стеки, черги та черги з пріоритетами.
7	Ввід/вивід в STL. Потoki вводу/виводу, маніпулятори.
8	Ітератори. Універсальний підхід для роботи з колекціями. Класи ітераторів. Зворотні ітератори. Ітератори вставки. Ітератори потоків вводу/виводу.
9	Робота з типами. Утиліти типів та числові ліміти.
10	Алгоритми в STL, ч.1. Немодифікуючі та модифікуючі алгоритми.
11	Алгоритми в STL, ч.2. Алгоритми видалення, перестановки, сортування. Алгоритми для впорядкованих діапазонів. Чисельні алгоритми.
Тема 3. Додаткові можливості стандартної бібліотеки C++.	
12	«Розумні» вказівники. Контроль динамічної пам'яті.
13	Регулярні вирази. Розширені можливості для роботи з рядками.
14	Додаткові можливості. Робота з часом та генератори псевдовипадкових чисел. Робота з файловою системою.
15	Багатопоточність засобами STL. Асинхронні виклики, потоки, засоби синхронізації, атомарні типи.
Тема 4. Використання сторонніх бібліотек.	
16	Огляд можливостей фреймворку QT. Робота з рядками, контейнери, файлами. Механізм слотів та сигналів.
17	Використання допоміжних бібліотек. Бібліотека Boost та інші бібліотеки.

18	Огляд можливостей C++/CLI. Задачі та принципи C++/CLI. Особливості роботи з C++/CLI.
----	---

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	Вирішення задач із використанням патернів проектування Завдання: Реалізувати класи, що вирішує поставлену згідно з варіантом задачу із використанням зазначених патернів проектування. Результати роботи оформити у вигляді проекту C++-програми та звіту.	6
2	Проектування класів з використанням стандартної бібліотеки C++. Завдання: Реалізувати класи, що вирішує поставлену згідно з варіантом задачу із використанням можливостей STL. Результати роботи оформити у вигляді проекту C++-програми та звіту.	4
3	Реалізація класу документу з використанням спеціальної мови розмітки Завдання: Реалізувати клас документу із можливістю конструювання за допомогою спеціальної мови розмітки та підтримкою виводу на екран. Результати роботи оформити у вигляді проекту C++-програми та звіту.	4
4	Реалізація багатопоточного додатку із використанням можливостей STL. Завдання: Відповідно до варіанту виіршити поставлену задачу із використанням механізмів багатопоточності, представлених в STL. 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	Ознайомлення з основами UML. Діаграми класів.	2
2	Знайомство з іншими патернами та архітектурними рішеннями. Типові рішення для різних задач.	2
3	Функціональні адаптери та прив'язки. Знайомство з std::bind.	2
4	Шаблон класу valarray. Застосування та основні можливості.	2
5	Шаблон класу bitset. Застосування та основні можливості.	2
6	Інтернаціоналізація вводу/виводу. Кодування, набори символів, локальні контексти та факети.	2

7	Робота з комплексними числами в STL. Використання шаблону <code>std::complex</code> .	2
8	Виключення в STL. Ознайомлення з ієрархією класів виключень.	2
9	Допоміжні шаблони. Ознайомлення з шаблонами <code>reference_wrapper</code> , <code>variant</code> , <code>any</code> , <code>optional</code> , <code>function</code> .	2
10	Засоби для роботи з файовою системою. Детальне ознайомлення з <code>std::filesystem</code> .	2
11	Тестування алгоритмів, ітераторів та адаптерів контейнерів. Використання ітераторів для вводу/виводу, обходу та наповнення колекцій.	2
12	Практичне використання «розумних» вказівників. Використання <code>weak_ptr</code> для організації циклічних зв'язків.	2
13	Розподілювачі пам'яті. Особливості використання власних розподілювачів пам'яті.	2
14	Практичне застосування бібліотеки регулярних виразів STL. Використання механізму регулярних виразів для задач обробки тексту.	2
15	Ознайомлення з Qt Creator. Реалізація простого проекту на основі Qt.	2
16	Реалізація GUI-застосунку із використанням Qt. Практичне застосування механізму слотів та сигналів. Реалізація власних класів із слотами та сигналами.	2
17	Використання засобів бібліотеки Boost. Практичне ознайомлення із можливостями бібліотеки Boost, підключення до проекту та використання.	2
18	Тестування механізму C++/CLI. Реалізація простого застосунку із використанням C++/CLI.	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-диск 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 англомовних 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.

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. заохочувальних та штрафних балів.

Лабораторні роботи

Загальна кількість балів – 64. Оцінка:

- «відмінно», повна відповідь на питання під час захисту (не менш ніж 90% потрібної інформації) та оформлений належним чином електронний протокол до лабораторної роботи – 15-16 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 – 12-14 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 – 10-11 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 – 5-9 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.

Модульна контрольна робота

Загальна кількість балів – 36. Модульна контрольна робота містить два теоретичних запитання та одне практичне завдання. Кожне запитання (завдання) оцінюється у 12 балів. Перелік теоретичних питань наведений у додатках.

Оцінка:

«відмінно», повна відповідь (не менш ніж 90% потрібної інформації) – 33-36 балів;

«добре», достатньо повна відповідь (не менш ніж 75% потрібної інформації), або повна відповідь з незначними помилками – 27-32 бали;

«задовільно», неповна відповідь (але не менш ніж 60% потрібної інформації) та незначні помилки – 22-27 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 – 11-21 бал;

«незадовільно» - 0-10 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 – 16 (1 лабораторна робота). На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менший ніж 6 балів.

За результатами 14 тижнів навчання максимально можлива кількість балів – 84 бали (3 лабораторні + МКР. На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менший ніж 42 бали.

Максимальна сума вагових балів контрольних заходів протягом семестру складає:

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

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

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

r_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 А. 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:

<i>Кількість балів</i>	<i>Оцінка</i>
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 А.

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

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

<https://www.coursera.org/learn/design-patterns>

<https://www.coursera.org/learn/cpp-basic-structures-vectors-pointers-strings-and-files>

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

<https://www.udemy.com/course/design-patterns-c>

<https://www.udemy.com/course/design-patterns-through-cpp>

<https://www.udemy.com/course/patterns-cplusplus>

<https://www.udemy.com/course/c-stl-complete-course>

<https://www.udemy.com/course/master-c-standard-template-library-stl>

<https://www.udemy.com/course/learn-intermediate-modern-c>

<https://www.udemy.com/course/data-structures-algorithms-using-c-zero-to-mastery>

<https://www.udemy.com/course/c-standard-template-library-in-practice>

<https://www.udemy.com/course/cplusplus-multithreading>

<https://www.udemy.com/course/multithreading-in-modern-c-c11-c14-c17>

<https://www.udemy.com/course/modern-cpp-concurrency-in-depth>
<https://www.udemy.com/course/learn-modern-cplusplus-concurrency>
https://www.udemy.com/course/multithreading_parta

а також інших, що відповідають тематиці курсу.

Робочу програму навчальної дисципліни (силабус):

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

Ухвалено кафедрою ____СПСКС____ (протокол № _11_ від _12.06.2024____)

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

Додаток А

Перелік теоретичних питань на залік

1. Патерни проектування: призначення, класифікація, характеристика.
2. Породжувальні патерни.
3. Структурні патерни.
4. Поведінкові патерни.
5. Пари та кортежі в STL.
6. Предикати та допоміжні функції в STL.
7. Послідовні контейнери в STL.
8. Впорядковані та невлпорядковані контейнери в STL.
9. Рядки в STL.
10. Адаптери контейнерів в STL.
11. Потоки вводу/виводу в STL.
12. Маніпулятори вводу/виводу в STL.
13. Ітератори в STL.
14. Утиліти типів в STL.
15. Алгоритми в STL. Призначення, класифікація, особливості використання.
16. Модифікуючі алгоритми в STL.
17. Немодифікуючі алгоритми в STL.
18. Алгоритми видалення, перестановки, сортування в STL.
19. Алгоритми для впорядкованих діапазонів та чисельні алгоритми в STL.
20. Розумні вказівники.

21. Регулярні вирази. Призначення та особливості. Бібліотека регулярних виразів в STL.
22. Реалізація багатопоточності в STL. Основні поняття та підходи.
23. Асинхронні виклики та потоки виконання.
24. Механізми синхронізації в STL. Загальна характеристика.
25. Механізм блокувань в STL.
26. Умовні змінні в STL.
27. Атомарні типи в STL.
28. Фреймворк QT. Загальна характеристика. Механізм слотів та сигналів. Призначення та особливості.
29. Бібліотека Boost. Призначення та особливості використання.
30. C++/CLI і .NET framework. Призначення та особливості.