

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА»



Факультет/інститут фізико-технічний

Кафедра комп'ютерної інженерії і електроніки

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Програмування. Частина 2

Освітня програма Комп'ютерна інженерія

Спеціалізація (за наявності) _____

Спеціальність 123 Комп'ютерна інженерія

Галузь знань 12 Інформаційні технології

Затверджено на засіданні кафедри
Протокол № __ від “_” ___ 2021 р.

м. Івано-Франківськ – 2021 р.

Зміст

1. Загальна інформація	3
2. Опис дисципліни	3
3. Структура курсу	5
4. Система оцінювання курсу	8
5. Оцінювання відповідно до графіку навчального процесу	8
6. Ресурсне забезпечення	8
7. Контактна інформація	9
8. Політика навчальної дисципліни	10

1. Загальна інформація

Назва дисципліни	Програмування. Частина 2.
Освітня програма	Комп'ютерна інженерія
Спеціалізація (за наявності)	
Спеціальність	123 Комп'ютерна інженерія
Галузь знань	12 Інформаційні системи
Освітній рівень	бакалавр
Статус дисципліни	Вибіркова
Курс / семестр	2 курс, 2 семестр
Розподіл за видами занять та годинами навчання (якщо передбачені інші види, додати)	Лекції – 32 год. Лабораторні заняття – 32 год. Самостійна робота – 116 год.
Мова викладання	Українська
Посилання на сайт дистанційного навчання	https://d-learn.pnu.edu.ua

2. Опис дисципліни

Мета та цілі курсу:	<p>Навчитися писати на C ++ ефективний код, вивчити ідіоми мови C ++, тобто показати, як за допомогою різних можливостей мови створювати елегантні, ефективні і надійні блоки коду. У сукупності зі знаннями, отриманими в основному курсі «Програмування», це дозволить студенту створювати не тільки швидкі і легко підтримувані програми. Крім того, будуть висвітлені теми, які не містилися в основному курсі «Програмування»: простори імен, розподіл пам'яті, багатопоточність і ефективне використання асоціативних контейнерів.</p> <p><i>Для вибору курсу та його опанування, необхідна підготовка, що відповідає знанням та вмінням з основного курсу «Програмування».</i></p>
Компетентності:	<p>Інтегральна компетентність: Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов.</p> <p>Загальні компетентності: ЗК1. Здатність до абстрактного мислення, аналізу і синтезу. ЗК2. Здатність вчитися і оволодівати сучасними знаннями.</p>

	<p>ЗК3. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК7. Вміння виявляти, ставити та вирішувати проблеми.</p> <p>Спеціальні (фахові компетентності):</p> <p>Р2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.</p> <p>Р3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.</p> <p>Р13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.</p> <p>Р15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.</p> <p>Здатність ефективно застосовувати у програмному коді на С++ макроси, розподіл пам'яті, лінійні контейнери, move-семантику, багатопоточність та асоціативні контейнери.</p>
<p>Програмні результати навчання:</p>	<p>Знання:</p> <p>Н3. Знати новітні технології в галузі комп'ютерної інженерії.</p> <p>Уміння:</p> <p>Н6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.</p> <p>Н8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.</p> <p>Н10. Вміти розробляти програмне забезпечення для вбудованих і розподілених додатків, гібридних систем.</p> <p>Н16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.</p>

3. Структура курсу

№	Тема	Результати навчання	Завдання
1	Макроси і шаблони класів.	Розуміти призначення макросів, знати випадки їх застосування та випадки, коли їх використовувати неприйнятно. Вміти програмувати макроси. Розуміти призначення шаблонів класів та навчитися їх створювати для розробки ефективного програмного забезпечення.	Бліц-питання на лекції, тести (макроси, вступ у шаблони класів, автоматичний вивід шаблонних типів), завдання для самостійної роботи
2	Принципи оптимізації коду, складність алгоритмів і ефективне використання вводу/виводу	Знати правила оптимізації коду та ефективно їх застосовувати на практиці. Вміти ефективно використовувати потоки вводу та виводу. Вміти оцінювати складність алгоритмів.	Бліц-питання на лекції, тест (складність алгоритмів), завдання для самостійної роботи
3	Модель пам'яті в C++	Розуміти як розподіляється пам'ять між різними об'єктами у C++, як її виділяти об'єкту, як звільняти виділену пам'ять. Навчитися ефективно керувати наявними ресурсами комп'ютерної системи у програмному коді.	Бліц-опитування під час викладу матеріалу, завдання, завдання для самостійної роботи
4	Ефективне використання лінійних контейнерів	Вміти ефективно використовувати контейнери вектору, дека, списку, масивів. Вивчення класу <code>string_view</code> та особливості його застосування.	Бліц-опитування, тести (вектори дек, списки, масиви і <code>string_view</code>), завдання для самостійної роботи
5	Move-семантика і базова багатопотоковість	Навчитися працювати з тимчасовими об'єктами, визначати ситуації коли переміщення об'єктів не ефективне, вміти створювати і застосовувати конструктори копіювання та оператори присвоювання, знати небезпеки оператора <code>return</code> . Вміти розпаралелювати задачі на	Бліц-питання, тести (переміщення і функція <code>move</code> , переміщення у стандартній бібліотеці, <code>move-ітератри</code> , <code>copy elision</code> і <code>NRVO</code>), завдання для

№	Тема	Результати навчання	Завдання
		<p>потоки для ефективного використання сучасних процесорів у прикладному програмному забезпеченні.</p>	<p>самостійної роботи</p>
6	<p>Ефективне використання асоціативних контейнерів</p>	<p>Знати та вміти використовувати асоціативні контейнери, хеш-функції, рекомендації з вибору хеш-функції, взаємозв'язок відсортованості контейнерів та продуктивності програмного коду, ітератори для роботи з контейнерами</p>	<p>Бліц-питання на лекції, тести (внутрішня будова асоціативних контейнерів, ітератори), завдання на самостійну роботу</p>
7	<p>Простори імен і вказівник <code>this</code></p>	<p>Розуміти мету та призначення простору імен, виявляти проблеми та їх уникати при перетині імен різних бібліотек, знати особливості синтаксису простору імен, особливості застосування глобального простору імен, директива <code>using namespace</code>, вміти структурувати програмний код із використанням простору імен, знати рекомендації з використання простору імен. Знати і розуміти призначення вказівника <code>this</code> та вміти його використовувати у програмному коді.</p>	<p>Бліц-питання під час лекції, тест (синтаксис роботи з просторами імен), завдання на самостійну роботу</p>
8	<p>Константність і <code>unique_ptr</code></p>	<p>Вміти використовувати <code>const</code> для підтримки інваріантів в класах і об'єктах, застосовувати константні об'єкти у багатопотокових програмах. Знати рекомендації з використання константності. Знати призначення розумних вказівників та вміти їх застосовувати при написанні програмного коду. Вміти виявляти витік пам'яті та</p>	<p>Бліц-питання під час лекції, тест (тест на константність,), завдання на самостійну роботу</p>

№	Тема	Результати навчання	Завдання
		застосовувати <code>unique_ptr</code> для виправлення витоку.	
9	Shared_ptr і RAII	Знати внутрішню будову розумних вказівників, розуміти і знати об'єкти з точки зору тривалості їх життя, розуміти суть поняття «володіти» об'єктом, вміти виявляти втрати пам'яті при застосуванні циклічних посилань і усувати такі проблеми, вміти застосовувати <code>shared_ptr</code> у багатопотокових програмах. Суть ідіоми RAII та її застосування.	Бліц-питання під час лекції, тест (тест на ідіома RAII), завдання на самостійну роботу
10	Функції: принципи зрозумілого коду	Розуміти суть застосування функцій, якими повинні бути функції, вміти передавати об'єкт та об'єкти у функцію, повернути об'єкт та групу об'єктів з функції	Бліц-питання під час лекції, тести (принципи зрозумілого коду, передача параметрів і повернення значень із функцій, ідея рефакторингу функцій), завдання на самостійну роботу

4. Система оцінювання курсу

Накопичування балів під час вивчення дисципліни	
Види навчальної роботи	Максимальна кількість балів
Лекції	10
Лабораторні роботи	70
Самостійна робота	20
Індивідуальне завдання	-
Залік	100
Максимальна кількість балів	100

5. Оцінювання відповідно до графіку навчального процесу

Види навчальної роботи	Навчальні тижні																					Разом	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21		
Лекції (ваг.коэф. 0,05)								100									100						10
Лабораторні роботи (ваг.коэф. 0,7)	100	100	100	100	100	100			100	100	100	100	100	100	100	100		100	100				70
Самостійна робота							10										10						20
Індивідуальні завдання																							
Залік (сума балів за всіма видами контролю)																							100
Всього за тиждень	100	100	100	100	100	100	10	100	100	100	100	100	100	100	100	100	110	100	100				100

6. Ресурсне забезпечення

Матеріально-технічне забезпечення	Мультимедійний проектор, комп'ютерна лабораторія для лабораторних робіт з IDE для програмування на C++ з кількістю комп'ютерів від 12 до 14, комп'ютерів
Література:	
1. Запухляк Р.І. Програмування на C++. –Івано-Франківськ: ВДЦ Прикарпатського національного університету імені Василя Стефаника,	

2009. -439 с. (мережа Інтернет).
2. Стивен Прата. Язык программирования С++ (С++11). Лекции и упражнения, 6-е издание — М.: Вильямс, 2012. — 1248 с.
 3. Б. Страуструп Язык программирования С++. Специальное издание. Пер. с англ. — М.: Издательство Бином, 2011 г. — 1136 с.
 4. Мейерс С. Эффективное использование С++. 35 новых рекомендаций по улучшению ваших программ и проектов. — М.: ДМК Пресс, 2014. — 294с.
 5. Мейерс С. Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ. — М.: ДМК Пресс, 2006. — 300с.
 6. Br. David Carlson. Software Design Using C++ (<https://cis.stvincent.edu/html/tutorials/swd/>)(інтернет).
 7. Pat Morin. Open Data Structures (in C++) (<https://docs.google.com/viewer?url=http%3A%2F%2Fopendatastructures.org%2Fods-cpp.pdf>)(інтернет).
 8. Скотт Мейерс. Эффективный и современный С++: 42 рекомендации по использованию С++11 и С++14. –Диалектика-Вильямс, 2020. – с. 304.
 9. Галовиц Я. С++17 STL. Стандартная библиотека шаблонов. — СПб.: Питер, 2018. — 432 с.
 - 10.Роберт С. Мартин. Гибкая разработка программ на Java и С++: принципы, паттерны и методики. –Издательство : Диалектика-Вильямс, 2020. –704 с.
 - 11.Страуструп Б. Дизайн и эволюция С++: Пер. с англ. – М.: ДМК Пресс, 2006. – 448 с.
 - 12.Энтони Уильямс. Параллельное программирование на С++ в действии. Практика разработки многопоточных программ. Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2012. – 672с.
 - 13.Бегун А.В. Технологія програмування: об'єктно-орієнтований підхід.- К.:КНЕУ,2000 .-200 с.(бібліотека 3 прим.).


7. Контактна інформація

Кафедра	Комп'ютерної інженерії та електроніки, вул. Шевченка, 57, 210а, 59-60-07, https://kkite.pnu.edu.ua/ , kkie@pnu.edu.ua
Викладач (і) Гостьові лектори	Запукхляк Руслан Ігорович, к.ф.-м.н., доцент
Контактна інформація викладача	ruslan.zapukhlyak@pnu.edu.ua

8. Політика навчальної дисципліни

Академічна доброчесність	<p>Студент повинен бути толерантним і поважати думку інших. Заперечення повинні формулюватися тільки в коректній формі. Плагіат та академічна недоброчесність несумісні з принципами діяльності ЗВО. Не допускається підказування та списування під час здачі будь-яких робіт поточного, рубіжного чи підсумкового контролю. Не допускається користування телефонами та будь-якими іншими електронними засобами під час здачі будь-яких робіт поточного, рубіжного, чи підсумкового контролю.</p> <p>За недотримання академічної доброчесності, студент може бути недопущений до складання підсумкового контролю та відрахований з університету.</p>
Пропуски занять (відпрацювання)	-1 бал за кожне заняття
Виконання завдання пізніше встановленого терміну	-20% балів від отриманого результату
Невідповідна поведінка під час заняття	-50% балів від отриманого результату
Додаткові бали	Додається до 10 балів за вчасне виконання всіх видів контролю, відвідування всіх занять, належну поведінку. Якщо у підсумку кількість балів є більшою за 100 балів, то підсумкова оцінка встановлюється така, що дорівнює 100.
Неформальна освіта	Можливість зарахування. Рекомендовані платформи: Coursera , Prometheus .

Викладач



Р.І. Запукхляк