Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра обчислювальної техніки

СИСТЕМНЕ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни	
Рівень вищої освіти	***Перший (бакалаврський)***
Галузь знань	*12 Інформаційні технології*
Спеціальність	*123 Комп'ютерна інженерія*
Освітня програма	*Комп'ютерні системи та мережі*
Статус дисципліни	*Нормативна*
Форма навчання	*Очна (денна)*
Рік підготовки, семестр	*2 курс, весняний семестр*
Обсяг дисципліни	*5 кредитів,150 годин з них лекцій 36 годин, лабораторних 18 годин, самостійна робота 96 годин*
Семестровий контроль/ контрольні заходи	*Екзамен*
Розклад занять	http://rozklad.kpi.ua/
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	*Доцент кафедри ОТ, кандидат техн. наук, Порєв Віктор Миколайович v_porev@ukr.net*
Розміщення курсу	Лекції: https://drive.google.com/drive/folders/1yJUO8ALYikkR-RyJlegD6iEuEix52CGI?usp=sharing Лабораторні роботи: https://drive.google.com/drive/folders/136iMz4R1lCSeYl8rOPldsigWXxGZYKB9?usp=sharing

Програма навчальної дисципліни

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

Метою вивчення дисципліни є формування у студентів здатностей:

- розуміти принципи керування процесором на програмному рівні;
- розуміти послідовність дій при створенні системних програм та впродовж життєвого циклу програмного забезпечення;
- розробляти системне програмне забезпечення за допомогою сучасних інструментальних засобів розробки програмного забезпечення;
- аналізувати процеси, які здійснюються під час компіляції, та знаходити можливі помилки;
- налагоджувати системні програми;

Основні завдання при вивченні дисципліни

Дисципліна «Системне програмування» забезпечує наступні програмні компетентності і програмні результати освітньо-професійної програми (ОПП): ФК2, ФК8, ФК12, ФК18, ПРН2, ПРН8, ПРН9, ПРН24

Згідно з вимогами ОПП здобувачі після засвоєння дисципліни « Системне програмування» мають продемонструвати такі програмні результати навчання:

- здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення
- здатність розв’язувати складні задачі і проблеми, що виникають у професійній діяльності.

знання:

- знання мови програмування Асемблер;
- засобів розробки програм на Асемблері;
- базового середовища виконання програм як у цілому, так і окремих елементів;
- команд, типів та форматів даних;
- методів забезпечення структурованості, модульності програм;
- взаємодії системних програм під час їх виконання;
- особливостей та етапів розробки системних програм;

уміння:

- розробляти та налагоджувати системні програми;
- розробляти та ефективно реалізовувати алгоритми, знаходити можливості оптимізації програмного коду;
- забезпечувати взаємодію модулів на Асемблері з модулями, написаними на інших мовах програмування;
- використовувати потрібні інструментальні засоби та інтегровані середовища для вирішення завдань;
- застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп’ютерних систем та мереж для вирішення технічних задач спеціальності

досвід:

- розроблення системного програмного забезпечення;
- роботи з комп’ютерними інструментальними засобами та інтегрованими середовищами.

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

Для успішного оволодіння дисципліною "Системне програмування" відповідно до освітньої програми необхідно попередньо оволодіти знаннями з дисциплін: "Програмування", "Структури даних і алгоритми", "Дискретна математика", "Комп'ютерна логіка", "Архітектура комп’ютерів".

Компетентності, знання та вміння, отримані в рамках вивчення даної дисципліни, можуть бути застосовані для вивчення дисциплін "Інженерія програмного забезпечення", "Операційні системи", "Системне програмне забезпечення", "Комп'ютерні системи", а також для виконання завдань освітнього компонента "Курсова робота з системного програмування".

3 Зміст навчальної дисципліни

Перелік основних тем, що входять до програми вивчення дисципліни "Системне програмування":

Розділ 1. Базове середовище виконання програм.

Тема 1.1 Огляд середовища виконання програм
Тема 1.2. Представлення чисел у комп’ютері
Тема 1.3. Організація пам'яті
Тема 1.4. Регістри процесора
Тема 1.5. Операнди команд та способи адресації операндів
Тема 1.6. Стек

Розділ 2. Основи програмування на Асемблері

Тема 2.1. Структура вихідного тексту програми на Асемблері. Основні директиви
Тема 2.2. Базові типи даних в Асемблері
Тема 2.3. Огляд системи команд. Основні команди та їхнє використання

Розділ 3. Структурованість та модульність програм

Тема 3.1. Поняття структурованості та модульності програм
Тема 3.2. Макроси
Тема 3.3. Процедури
Тема 3.4. Модулі
Тема 3.5. Використання системних функцій та бібліотек

Розділ 4. Середовище x87 FPU

Тема 4.1. Особливості середовища x87 FPU
Тема 4.2. Основні команди x87 FPU та їхнє використання

Розділ 5. Огляд розширень архітектури процесорів та їхнє застосування

Тема 5.1. Розширення SIMD та їхні різновиди
Тема 5.2. Розширення SSE
Тема 5.3. AVX та інші розширення архітектури

Розділ 6. Асемблер і мови програмування високого рівня

Тема 6.1. Конвенції виклику
Тема 6.2. Використання модулів на Асемблері у проектах на мовах програмування високого рівня

4 Навчальні матеріали та ресурси

Базова література:

1. Лекції знаходяться за посиланням https://drive.google.com/drive/folders/1yJUO8ALYikkR-RyJlegD6iEuEix52CGI?usp=sharing
2. Системне програмування. Програмування на асемблері. Комп'ютерний практикум: Навч. посібник. [Електронний ресурс] / Уклад.: В.М. Порєв. – Електронні текстові дані (1 файл: 4,3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 106 с. Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол №1 від 02.09.2022р.) за поданням Вченої ради факультету інформатики та обчислювальної техніки (протокол №11 від 11.07.2022 р.) http://comsys.kpi.ua/
3. Актуальні завдання для лабораторних робіт знаходяться за посиланням https://drive.google.com/drive/folders/136iMz4R1lCSeYl8rOPldsigWXxGZYKB9?usp=sharing

Додаткова:

4. Intel® 64 and IA-32 Architectures. Software Developer's Manual. https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html
5. Microsoft Macro Assembler reference https://learn.microsoft.com/en-us/cpp/assembler/masm/microsoft-macro-assembler-reference?view=msvc-170
6. C and C++ in Visual Studio https://learn.microsoft.com/en-us/cpp/overview/visual-cpp-in-visual-studio?view=msvc-170

Обладнання, що необхідне для проведення занять

Лекційні заняття проводяться в аудиторії, яка обладнано проектором, практичні заняття – в комп'ютерному класі. Для дистанційних занять потрібен доступ до інтернет та програмне й апаратне забезпечення для проведення відеоконференцій.

Навчальний контент

5 Методика опанування навчальної дисципліни (освітнього компонента)

5.1 Лекційні заняття

№ з/п	Назва теми (тем) лекційного заняття та перелік основних питань
1	**Тема 1.1. Огляд середовища виконання програм** Поняття базового середовища виконання програм. Режими роботи процесора архітектури IA-32. Елементи середовища, які надає архітектура IA-32. Основні відмінності 64-бітного середовища від 32-бітного **Завдання на СРС:** – Різновиди та приклади сучасних інтерпретаторів та компіляторів. Інтегровані середовища розробки програм – Програмна реалізація обробників повідомлень – Особливості програмування додатків для Windows та Android – Приклади API
2	**Тема 1.2. Представлення чисел у комп'ютері** Позиційні системи числення. Переведення чисел в іншу систему. Представлення цілих чисел зі знаком. Представлення дробових чисел. Стандарт IEEE 754 та його розвиток. Стандартні формати з плаваючою точкою **Завдання на СРС:** – Доповняльний код – Двійкові формати Single, Double Precision
3	**Тема 1.3. Організація пам'яті** Фізична пам'ять. Сегментація пам'яті та сторінкова адресація. Сторінкова адресація та віртуальна пам'ять. Моделі пам'яті. Модель реальної адресації. Модель сегментованої пам'яті. Модель flat **Завдання на СРС:** – Особливості віртуальної памяті для 32- та 64-бітових архітектур **Тема 1.4. Регістри процесора** Регістри процесора та їхні різновиди. Регістри загального призначення в архітектурі IA -32.

	Сегментні регістри. Регістр EFLAGS **Завдання на СРС:** - Відмінності регістрів 32-бітових та 64-бітових архітектур Intel
4	**Тема 1.5. Операнди команд та способи адресації операндів** Безпосередні операнди. Регістрові операнди. Операнд пам'ять. Вказування селектора сегменту у програмах на Асемблері. Вказування зсуву. Пряма та опосередкована адресація. Вказування зсуву. Пряма адресація. Зміщення. Вказування зсуву. Опосередкована адресація. База + Індекс х Масштаб + Зміщення **Завдання на СРС:** – Базові типи даних – Розташування даних у пам'яті. Вирівнювання – Числові типи даних. Вказівники
5	**Тема 1.6. Стек** Поняття стеку та особливості його організації. Доступ до стеку. Вказівник стеку. Команди PUSH, POP. **Завдання на СРС:** - Можливості керування розміром стеку для програм - Відмінність організації стеку у 32- та 64-бітових архітектурах **Тема 2.4. Огляд системи команд** Огляд системи команд. Основні групи команд. Команди MOV, ADD, ADC, SUB, SBB. **Завдання на СРС:** - Особливості використання переносу при додаванні та відніманні даних підвищеної розрядності
6	**Тема 2.4. Огляд системи команд** Команди переходів. Команда JMP. Команди умовних переходів Jcc. Програмування циклів. Програмування вкладених циклів. Команди LOOP,LOOPE,LOOPNE **Завдання на СРС:** – Програмування циклів з постумовою та передумовою
7	**Тема 2.4. Огляд системи команд** Команди множення та ділення. Команди MUL, IMUL, DIV, IDIV. Різновиди форматів та операнди команд множення та ділення. Переповнення та інші особливі ситуації виконання **Завдання на СРС:** - Можливості організації множення та ділення даних підвищеної розрядності
8	**Тема 2.4. Огляд системи команд** Побітові логічні команди. Команди зсувів. Запис та читання окремих бітів **Завдання на СРС:** - Використання побітових команд та зсувів при обробці бітових полів довільного розміруі
9	**Тема 2.4. Огляд системи команд** Команди обробки рядкових даних. Команди STOS, STOSx. Команди MOVS, MOVSx. Команди CMPS, CMPSx. Команди SCAS, SCASx. Префікси REP, REPE, REPNE, REPZ, REPNZ **Завдання на СРС:** – Використання STOS для швидкої ініціалізації буферів – Вирішення проблеми перекриття при копіюванні MOVS – Особливості використання префіксів REPx
10	**Тема 3.1. Поняття структурованості та модульності програм** Огляд можливостей забезпечення структурованості та модульності програм на Асемблері. **Завдання на СРС:** - Складання модульних проектів на Асемблері у середовищі Microsoft Visual Studio

	– Роздільна компіляція та налагодження програмних модулів **Тема 3.2. Макроси** Поняття макросу, формат визначення. Макроси з параметрами. Особливості використання міток у макросах **Завдання на СРС:** – Синтаксис макросів в MASM32
11	**Тема 3.2. Процедури** Поняття процедури. Визначення, виклик, способи передачі параметрів. Передача параметрів через стек. Перемінна кількість параметрів. Вказування параметрів процедур для компілятора **Завдання на СРС:** – Порівняння швидкодії використання процедур для різних способів передавання параметрів
12	**Тема 3.2. Процедури** Локальні дані процедур. Пролог та епілог процедури. Стековий кадр (фрейм). Стекові кадри при вкладених викликах. Рекурсія. Проблема переповнення стеку. **Завдання на СРС:** – Особливості програмування локальних даних – Можливості контролю стека при налагодженні програм
13	**Тема 4.1. Особливості середовища x87 FPU** – Середовище x87 FPU. Стек даних x87 FPU. Типи даних x87 FPU. Завантаження даних у стек x87 FPU та читання зі стеку **Завдання на СРС:** – Перетворення форматів даних з плаваючою точкою **Тема 4.2. Основні команди x87 FPU та їхнє використання** Огляд команд x87 FPU. Приклад обчислень у середовищі x87 FPU **Завдання на СРС:** – Обчислення скалярного добутку з довільною кількістю елементів – Налагодження програмного коду x87 FPU у середовищі розробки на асемблері
14	**Тема 5.1. Розширення SIMD та їхні різновиди** Розширення SIMD. Основні особливості. Технологія MMX. Приклади команд **Завдання на СРС:** – Використання команд MMX у програмному забезпеченні мультимедіа
15	**Тема 5.2. Розширення SSE** Подальший розвиток SSE: SSE2, SSE3, SSSE3, SSE4. Горизонтальне та вертикальне додавання **Завдання на СРС:** – Векторні команди SSE обробки чисел у форматах з плаваючою точкою – Цілочисельні команди SSE
16	**Тема 5.3. AVX та інші розширення архітектури** Подальший розвиток SIMD: AVX. Регістри AVX і формати даних. Відмінності команд AVX від SSE. Команди AVX - спадкоємиці команд SSE. Нові команди AVX. Особливості використання команд AVX. Перевірка підтримки AVX **Завдання на СРС:** – Команди векторних операцій – Горизонтальне та вертикальне додавання AVX – Команди шифрування даних
17	**Тема 6.1. Конвенції виклику** Асемблер і мови високого рівня. Поняття конвенції виклику. Огляд конвенцій виклику.

	Способи передачі параметрів та результату у різних конвенціях виклику. Конвенція cdecl. Конвенція stdcall. **Завдання на СРС:** – Написання процедур згідно конвенції cdecl для проектів Visual Studio C++ – Написання процедур згідно конвенції stdcall для проектів Visual Studio C++ **Тема 6.2. Використання модулів на Асемблері у проектах на мовах програмування високого рівня.** Огляд засобів розробки програмного забезпечення з підтримкою ассемблера. Підключення модулів на асемблері у проекти C++. Асемблерні вставки. **Завдання на СРС:** – Створення та підключення модулів на Асемблері у середовищі Microsoft Visual Studio
18	**Модульна контрольна робота** Питання за роділами 1-6

5.2 Лабораторні роботи

№ з/п	Назва лабораторної роботи та перелік основних питань	Кількість ауд. годин
1	**Лабораторна робота №1. Знайомство із середовищем розробки програм на Асемблері** Мета: Отримати перші навички роботи з середовищем розробки для створення програм, написаних мовою Асемблера. Для виконання роботи потрібно вивчення окремих відомостей за наступними темами: – Тема 2.1. Структура вихідного тексту програми на Асемблері. Основні директиви – Тема 2.2. Базові типи даних в Асемблері – Тема 2.3. Огляд системи команд. Основні команди та їхнє використання Потрібно навчитися створювати проекти, писати та компілювати вихідні тексти, налагоджувати програми на Асемблері з використанням засобів MASM32, Microsoft Visual Studio	**2**
2	**Лабораторна робота №2. Створення модульних проектів на Асемблері та вивчення форматів представлення чисел** Для виконання роботи потрібно вивчення окремих відомостей за наступними темами: – Тема 1.2. Представлення чисел у комп'ютері – Тема 2.1. Структура вихідного тексту програми на Асемблері. Основні директиви – Тема 2.2. Базові типи даних в Асемблері Потрібно опанувати питання щодо представлення чисел у різних системах числення та їхнє кодування у стандартних форматах. Зокрема, вивчається можливості програмування цілих чисел у доповняльному коді та у форматах з плаваючою точкою різної довжини. Отримуються вміння та навички використання базових типів даних процессорів.	**2**
3	**Лабораторна робота №3. Програмування арифметичних операцій підвищеної розрядності** Мета: Навчитися програмувати на асемблері основні арифметичні операції підвищеної розрядності, а також отримати перші навички програмування власних процедур у модульному проекті.	**2**

	Для виконання роботи потрібно вивчення окремих відомостей за наступними темами: – Тема 1.1 Огляд середовища виконання програм – Тема 1.2. Представлення чисел у комп'ютері – Тема 1.4. Регістри процесора – Тема 2.3. Огляд системи команд. Основні команди та їхнє використання – Тема 3.3. Процедури – Тема 3.4. Модулі Потрібно ознайомитися з особливостями додавання та віднімання підвищеної розрядності, вивчити особливості команд ADD, ADC, SUB та SBB. Отримати вміння написання процедур на Асемблері та складати модульні проекти.	
4	**Лабораторна робота №4. Програмування множення чисел підвищеної розрядності** Мета: Навчитися програмувати на асемблері множення чисел підвищеної розрядності, а також закріпити навички програмування власних процедур у модульному проекті. Для виконання роботи потрібно вивчення окремих відомостей за наступними темами: – Тема 1.1 Огляд середовища виконання програм – Тема 1.2. Представлення чисел у комп'ютері – Тема 1.4. Регістри процесора – Тема 2.3. Огляд системи команд. Основні команди та їхнє використання – Тема 3.3. Процедури – Тема 3.4. Модулі Потрібно ознайомитися з алгоритмами множення підвищеної розрядності. Вивчити особливості команди MUL, програмування вкладених циклів. Навчитися оптимально використовувати наявні регістри.	**2**
5	**Лабораторна робота №5. Програмування побітових операцій** Мета: Навчитися програмувати на асемблері побітові операції, вивчити основні команди обробки бітів. Для виконання роботи потрібно вивчення окремих відомостей за темою: Тема 2.3. Огляд системи команд. Основні команди та їхнє використання. Потрібно ознайомитися з командами побітових операцій та зсувів. Запрограгрумати обробку бітових полів довільної розрядності.	**2**
6	**Лабораторна робота №6. Програмування операцій ділення чисел** Мета: Навчитися програмувати на асемблері ділення чисел, вивчити перетворення з двійкової у десяткову систему числення. Для виконання роботи потрібно вивчення окремих відомостей за наступними темами: – Тема 1.2. Представлення чисел у комп'ютері – Тема 2.3. Огляд системи команд. Основні команди та їхнє використання Потрібно ознайомитися з алгоритмами ділення чисел довільної розрядності. Запрограмувати два варіанти ділення на основі використання команд побітової обробки та команд DIV, IDIV.	**2**
7	**Лабораторна робота №7. Виконання операцій з плаваючою точкою та вивчення команд x87 FPU** Мета: Навчитися програмувати операції з плаваючою точкою на асемблері. Для виконання роботи потрібно вивчення окремих відомостей за наступними темами:	**2**

	– Тема 1.2. Представлення чисел у комп’ютері – Тема 4.1. Особливості середовища x87 FPU – Тема 4.2. Основні команди x87 FPU та їхнє використання Потрібно ознайомитися з побудовою підсистеми x87 FPU. Стек даних FPU. Основні команди виконання операцій математичних та завантаження даних. Вивчити стандартні формати представлення чисел з плаваючою точкою. Запрограмувати переведення двійкового представлення чисел з плаваючою точкою у десяткову систему числення.	
8	**Лабораторна робота №8. Використання функцій API у програмах на асемблері** Мета: Навчитися використовувати у програмах на асемблері функції Windows динамічного виділення пам’яті та запису файлів. Для виконання роботи потрібно вивчення окремих відомостей за наступними темою: Тема 3.5. Використання системних функцій та бібліотек Потрібно ознайомитися з Application Program Interface цільової операційної платформи. Запрограмувати створення динамічних масивів, запису файлів, стандартні діалогові вікна шляхом виклику системних функцій.	**2**
9	**Лабораторна робота №9. Використання у проекті C++ модулів на асемблері. Програмування команд SIMD у модулях на асемблері** Мета: Навчитися створювати програми на C++ з використанням модулів на асемблері. Навчитися програмувати модулі на асемблері, у яких містяться команди SSE, команди x87 FPU, а також використовувати такі модулі у проектах C++. Для виконання роботи потрібно вивчення окремих відомостей за наступними темами: – Тема 5.2. Розширення SSE – Тема 5.3. AVX та інші розширення архітектури – Тема 6.1. Конвенції виклику – Тема 6.2. Використання модулів на Асемблері у проектах на мовах програмування високого рівня Потрібно створити проект C++, запрограмувати графічний інтерфейс користувача та процедури на асемлбері в окремих модулях. Проаналізувати доцільність використання SIMD команд розширень архітектури сучасних процесорів.	**2**

1.3. Перелік питань на модульну контрольну роботу та на екзамен

1. Базове середовище виконання програм
2. Режими роботи процесора архітектури IA-32
3. Елементи середовища, які надає архітектура IA-32
4. Основні відмінності 64-бітного середовища від 32-бітного
5. Представлення чисел у комп'ютері
6. Позиційні системи числення
7. Переведення чисел в іншу систему
8. Представлення цілих чисел зі знаком
9. Представлення дробових чисел
10. Стандарт IEEE 754 та його розвиток
11. Стандартні формати з плаваючою точкою
12. Організація пам’яті
13. Фізична пам’ять

14. Сегментація пам'яті та сторінкова адресація
15. Сторінкова адресація та віртуальна пам’ять
16. Моделі пам'яті
17. Модель реальної адресації
18. Модель сегментованої пам’яті
19. Модель flat
20. Регістри процесора та їхні різновиди
21. Регістри загального призначення в архітектурі IA -32
22. Сегментні регістри
23. Регістр EFLAGS
24. Операнди команд та способи адресації операндів
25. Безпосередні операнди
26. Регістрові операнди
27. Операнд пам'ять
28. Вказування селектора сегменту у програмах на Асемблері
29. Вказування зсуву. Пряма та опосередкована адресація
30. Вказування зсуву. Пряма адресація. Зміщення
31. Вказування зсуву. Опосередкована адресація. База
32. Вказування зсуву. База + зміщення
33. Вказування зсуву. Індекс х Масштаб + Зміщення
34. Вказування зсуву. База + Індекс х Масштаб + Зміщення
35. Стек
36. Базові типи даних
37. Розташування даних у пам'яті. Вирівнювання
38. Числові типи даних. Вказівники
39. Огляд системи команд
40. Команда MOV
41. Команди ADD, ADC
42. Команди SUB, SBB
43. Команда MUL
44. Команда IMUL
45. Команда DIV
46. Команда IDIV
48. Побітові логічні команди
49. Команди зсувів
50. Запис та читання окремих бітів
51. Команди переходу
52. Команда JMP
53. Команди умовних переходів Jcc
54. Програмування циклів
55. Програмування вкладених циклів
56. Команди LOOP,LOOPE,LOOPNE
57. Команди обробки рядкових даних

58. Команди STOS, STOSx
59. Команди MOVS, MOVSx
60. Команди CMPS, CMPSx
61. Команди SCAS, SCASx
62. Префікси REP,REPE,REPNE,REPZ,REPNZ
63. Структурованість та модульність програм
64. Макроси
65. Процедури. Визначення,виклик,способи передачі параметрів
66. Процедури. Передача параметрів через стек
67. Процедури. Перемінна кількість параметрів
68. Вказування параметрів процедур для компілятора
69. Локальні дані процедур
70. Пролог та епілог процедури
71. Стековий кадр (фрейм)
72. Середовище x87 FPU
73. Стек даних x87 FPU
74. Типи даних x87 FPU
75. Огляд команд x87 FPU
76. Приклад обчислень у середовищі x87 FPU
77. Розширення SIMD. Основні особливості
78. MMX.Огляд
79. MMX. Приклади команд
80. Технологія SSE. Огляд
81. Команди векторних операцій SSE
82. Команди MOVAPS, MOVUPS
83. Подальший розвиток SSE: SSE2, SSE3, SSSE3, SSE4 ...
84. Горизонтальне та вертикальне додавання
85. Подальший розвиток SIMD: AVX
86. Регістри AVX і формати даних
87. Відмінності команд AVX від SSE
88. Команди AVX - спадкоємиці команд SSE
89. Нові команди AVX
90. Особливості використання команд AVX
91. Перевірка підтримки AVX
92. Асемблер і мови високого рівня
93. Поняття конвенції виклику
94. Огляд конвенцій виклику
95. Способи передачі параметрів та результату у різних конвенціях виклику
96. Конвенція cdecl
97. Конвенція stdcall
98. Написання процедур згідно конвенції cdecl
99. Написання процедур згідно конвенції stdcall
100. Текст найпростішої програми на Асемблері

101. Директиви Асемблера
102. Засоби розробки програм на Асемблері
103. Створення проекту на MASM32
104. Створення проекту на Visual Studio
105. Створення модульних програм на Асемблері
106. Використання у проекті C++ модулів на Асемблері
107. Налагодження програм та аналіз коду: точки зупинки,контроль значень регістрів, дизасемблер

6 .Самостійна робота студента (СРС)

Студенти повинні закріпляти знання, отримані під час лекцій та поглиблювати свої знання для подальшого навчання. Крім того, самостійна робота необхідна при виконанні лабораторних робіт та підготовки для екзамену.

Перелік завдань на СРС надано у відомостях п. 5.1. Лекційні заняття

Політика та контроль

7 Політика навчальної дисципліни (освітнього компонента)

Всі студенти повинні відвідувати лекційні та лабораторні заняття – як при звичайному навчанні у аудиторіях (фізичне відвідування), так і при дистанційному навчанні (віртуальне відвідування).

Результати виконання лабораторних робіт оформлюються у електронному форматі у вигляді файлів звітів, виконуваних файлів та інших файлів. Такі файли повинні містити результати відповідно завданням, вимогам та методичним вказівкам для кожної роботи. Робота програм перевіряється викладачем в ході прийому робіт.

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Чим більше запізнення, тем менша оцінка.

Усі письмові роботи перевіряються на наявність плагіату. Плагіат суттєво зменшує оцінку, причому значне запозичення чужого тексту може призвести до незадовільної оцінки роботи. Списування під час заходів контролю заборонені (в т. ч. із використанням мобільних пристроїв).

8 Види контролю та рейтингова система оцінювати результатів навчання (РСО)

Підсумкова рейтингова оцінка ($R_Д$) студента з дисципліни "Об'єктно-орієнтоване програмування" складається з балів, які він отримує:

- за 9 лабораторних робіт ($R_{ЛАБ}$);
- за модульну контрольну роботу ($R_{МКР}$);
- за екзамен (R_E).

Відповідно до "Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського" (https://osvita.kpi.ua/sites/default/files/downloads/Pol_systema_ociniuvannia.pdf), затвердженого Наказом №1/273 від 14.09.2020 р., застосована система оцінювання типу РСО-2, для якої рейтингова оцінка складається з:

- стартової оцінки (R_C) – оцінювання заходів впродовж семестру: $R_C = R_{ЛАБ} + R_{МКР}$,
- екзаменаційної оцінки R_E

Таким чином, $R_Д = R_C + R_E$

Розрахунок шкал оцінювання

Шкала оцінювання (максимально можлива оцінка) одної лабораторної роботи: 6 балів.
Шкала оцінювання 9 лабораторних робіт: ***RЛАБ*** = 9 × 6 = 54 балів
Шкала оцінювання модульної контрольної роботи: ***RМКР*** = 6 балів
Шкала стартової оцінки: ***RС*** = ***RЛАБ*** + ***RМКР*** = 60 балів
Шкала екзаменаційної оцінки: ***RE*** = 40 балів
Повна загальна шкала рейтингу: ***R*** = ***RС*** + ***RE*** = 60 + 40 = 100 балів
Мінімально можлива загальна позитивна оцінка: ***RМІН*** = 0.6 × 100 = 60 балів

Визначення умов допуску до екзамену

Умовою допуску до екзамену є виконання з певною успішністю завдань впродовж семестру, тобто, студент має впродовж семестру заробити деякий ненульовий стартовий рейтинг (***R_C*** > 0).

Мінімальний стартовий рейтинг для допуску до екзамену (***RМІНДОП***) визначається з припущення, що отримавши максимально можливу оцінку на екзамені (***RE***) студент в результаті буде мати мінімальну позитивну оцінку (***RМІН),*** тобто ***RМІН = RМІНДОП + RE.*** Звідти значення ***RМІНДОП*** = ***RМІН*** – ***RE*** = 60 – 40 = 20 балів. Таким чином, умовою допуску до екзамену буде ***R_C*** >= ***RМІНДОП,*** тобто, значення стартового рейтингу ***R_C*** має бути не менше 20 балів.

Таким чином, при умові допуску до екзамену, студент в результаті успішного складання екзамену буде мати підсумкову рейтингову оцінку (***$R_Д$***), яка буде сумою усіх отриманих оцінок у балах. Підсумкова рейтингова оцінка також переводиться в оцінку за університетською шкалою.

Таблиця переведення підсумкової рейтингової оцінки в університетську шкалу оцінок

Підсумкова рейтингова оцінка ***$R_Д$***	Університетська шкала оцінок
95...100	Відмінно
85...94	Дуже добре
75...84	Добре
65...74	Задовільно
60...64	Достатньо
Менше 60	Незадовільно
Стартовий рейтинг (***Rс***) менше 20	Не допущено

Робочу програму навчальної дисципліни (силабус):

Склав доцент кафедри обчислювальної техніки, к.т.н. Порєв В. М.

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25 травня 2022 р.)

Погоджено Методичною комісією факультету інформатики та обчислювальної техніки (протокол № 10 від 9 червня 2022 р.)