



# АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ:

## Частина 1. ОРГАНІЗАЦІЯ ПРОГРАМ МОВОЮ ВИСОКОГО РІВНЯ C++

### Робоча програма навчальної дисципліни (Силабус)

#### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>122 Комп'ютерні науки</i>
Освітня програма	<i>Комп'ютерні технології в біології та медицині</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна(денна)</i>
Рік підготовки, семестр	<i>1 курс, весняний семестр; 2 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити ЄКТС / 120 год. (28 год. – лекції, 26 год. – комп'ютерні практикуми, 66 год. – СР)</i>
Семестровий контроль/ контрольні заходи	<i>залік, модульна контрольна робота (МКР)</i>
Розклад занять	<i><a href="http://rozklad.kpi.ua">http://rozklad.kpi.ua</a></i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н. Алхімова Світлана Миколаївна, контактний телефон: +380674045083, e-mail: <a href="mailto:alkhimova.svitlana@lil.kpi.ua">alkhimova.svitlana@lil.kpi.ua</a></i>
Розміщення курсу	<i>Посилання на дистанційний ресурс Google classroom: <a href="https://classroom.google.com/c/NTg4Mzk5ODAxMzE4">https://classroom.google.com/c/NTg4Mzk5ODAxMzE4</a></i>

#### Програма навчальної дисципліни

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

Дисципліна «Алгоритмізація та програмування» складається з двох освітніх компонентів:

Частина 1 – Організація програм мовою високого рівня C++;

Частина 2 – Структури даних і алгоритми їх обробки.

**Предметом навчальної дисципліни** є система здатностей та умінь із основ алгоритмізації задач в предметній галузі та створення програмних застосунків мовою програмування C++, які необхідні під час виконання обов'язків, виробничих функцій та типових задач діяльності фахівця.

**Метою навчальної дисципліни** є засвоєння студентами необхідних знань з програмування мовою високого рівня C++, а також формування у студентів цілісного уявлення про розробку алгоритмів для вирішення задач у предметній галузі та базової підготовки в галузі використання засобів обчислювальної техніки, а також підготовки студентів до виконання науково-дослідної роботи на старших курсах на професійному рівні.

**Програмні результати навчання.**

Студенти після засвоєння навчальної дисципліни мають набути наступні компетентності.

##### Загальні компетентності

**ЗК 1** Здатність до абстрактного мислення, аналізу та синтезу

**ЗК 2** Здатність застосовувати знання у практичних ситуаціях

**ЗК 3** Знання та розуміння предметної області та розуміння професійної діяльності

**ЗК 6** Здатність вчитися і оволодівати сучасними знаннями

### **Фахові компетентності**

**ФК 3** Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

**ФК 8** Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

**ФК 18** Здатність проектувати та реалізовувати програмне забезпечення для вирішення різноманітних задач в біології та медицині із застосуванням різних мов програмування та сучасних бібліотек крос-платформного програмування, проводити тестування на декількох апаратних платформах, впроваджувати і підтримувати роботу інформаційних систем на сучасних платформах.

Студенти після засвоєння навчальної дисципліни мають продемонструвати наступні програмні результати навчання

**ПР 1** Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук

**ПР 5** Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

**ПР 9** Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

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

В структурно-логічній схемі програми підготовки фахівця навчальна дисципліна «Алгоритмізація та програмування. Частина 1-2» входить до переліку нормативних дисциплін, циклу професійної підготовки.

**Пререквізити.** Навчальна дисципліна «Алгоритмізація та програмування» базується на раніше засвоєній навчальній дисципліні «Основи інформатики та програмування».

**Постреквізити.** Навчальна дисципліна забезпечує наступну дисципліну «Об'єктно-орієнтоване програмування», а також є основою для підготовки дипломних робіт за спеціальністю та в подальшій практичній роботі за фахом.

## **3. Зміст навчальної дисципліни**

### **Освітній компонент 1. «Організація програм мовою високого рівня C++»**

#### **Тема 1. Основи алгоритмізації.**

Етапи розв'язування обчислювальних задач на комп'ютері. Поняття алгоритму, його властивості; найбільш поширені способи опису алгоритмів: словесний запис, графічні схеми алгоритмів (блок-схеми), псевдокод (формальні алгоритмічні мови). Базові алгоритмічні структури. Правила побудови та визначення блок-схем алгоритмів для написання програм на мові програмування C++, графічні елементами та базові структурами блок-схем.

#### **Тема 2. Позиційні системи числення**

Позиційні системи числення, особливості роботи з ними. Правила переведення чисел з однієї системи числення в іншу. Арифметичні операції (додавання, віднімання, множення, ділення) над числами у різних системах числення

**Тема 3.** Елементи алгоритмічних мов: поняття типу даних, ідентифікатори, змінні, константи, операції Історією розвитку і стандарт мови С++. Алфавіт і лексеми мови С++ (ідентифікатори, оператори, константи, ключові слова, роздільники). Структура програм, що написані мовою С++. Основні стадії створення виконуваного файлу; процеси компіляції та компонування. Створення проекту у Microsoft Visual Studio; механізми редагування, компіляції та запуску простої програми, написаної на С++ з використанням Microsoft Visual Studio. Зберігання інформації у пам'яті комп'ютера; числа з фіксованою крапкою; числа з рухомою крапкою; машинний нуль; машинна нескінченність. Поняття типу даних; стандартні типів мови програмування С++. Об'явлення, визначення та ініціалізації змінних; операції та вирази мови С++; введення / виведення даних, прості команди С++.

**Тема 4.** Структурне програмування: послідовність, розгалуження та цикли Умовний оператор if та умовний вираз; оператор вибору варіантів switch; оператори циклу for, while, do-while; оператори безумовної адресації break та continue. Визначення покажчика та посилання; операція взяття адресу &; операція розіменування; асоціативність операцій; особливості вживання модифікатора const з покажчиками.

**Тема 5.** Процедурно-орієнтоване програмування. Рекурсія Об'явлення (прототип), визначення та виклик функцій користувача; формальні та фактичні параметри функції; способи передачі параметрів у функцію; константні параметри функції; функція з параметрами за замовчуванням; механізм повернення значення із функції, оператор return; покажчики на функції; рекурсія. Глобальні та локальні змінні; область дії (видимості) змінних; час існування змінних; простір імен.

## 4. Навчальні матеріали та ресурси

### Базова

1. Бородкіна, І.Л. Теорія алгоритмів : посібник для студентів вищих навчальних закладів / І.Л. Бородкіна, Г.О. Бородкін. – К.: Центр навчальної літератури, 2019. – 184 с.
2. Ришковець, Ю. В., Висоцька, В. А. Алгоритмізація та програмування. Частина 1: навч. посіб. – Львів: «Новий Світ-2000», 2018. – 337 с.
3. С++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. – [2-ге вид. перероб. і доповн]. – Одеса : Фенікс, 2019. – 477 с.
4. Основи програмування мовою С++ [Текст] : навч. посіб. / Є. П. Путятін [та ін.]. - Харків : Коряк С. Ф., 2018. – 282 с.
5. Алгоритмізація та програмування мовою високого рівня С++ : Комп'ютерний практикум : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки та інформаційні технології», спеціалізації «Інформаційні технології в біології та медицині» / С. М. Алхімова. – К.: Вид-во «Політехніка», КПІ ім. Ігоря Сікорського, 2018. – 156 с.

### Допоміжна

1. Грязнова, В. О., Єфіменко, С. В. Основи методології програмування. – К. : ВПЦ "Київський університет", 2005. – 364 с.
2. Stroustrup, B. C++ Programming Language / Bjarne Stroustrup. – [3rd ed.]. – Boston, MA : Addison Wesley Professional, 2000. – 1030 p.
3. Stroustrup, B. Programming : principles and practice using C++ / Bjarne Stroustrup. – [2nd ed.]. – Boston, MA : Addison Wesley Professional, 2014. – 1312 p.
4. Мова С++ в програмуванні та комп'ютерних науках : навч. посіб. з курсів "Основи програмування", "Алгоритмізація та програмування" / В. Ю. Воловщиков [та ін.] ; під ред. проф. Годлевського М. Д. ; Нац. техн. ун-т "Харків. політехн. ін-т". – Харків : Мезіна В. В. [вид.], 2017. – 279 с.
5. С++. Теорія та практика: навч. посібник / [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.]; за ред. О. Г. Трофименко. – Одеса: ОНАЗ ім. О.С. Попова, 2011. – 588 с.
6. Татарчук, Д. Д., Діденко, Ю. В. Програмування мовами С та С++: навч. посіб. / Д.Д. Татарчук, Ю.В. Діденко. – К. : НТУУ «КПІ», 2012. – 112 с.
7. Путятін, Є. П. Основи програмування мовою С++ / Є. П. Путятін, В.П. Степанов, В. П. Пчелінов, Т. Г. Долженкова, О. О. Матат. – Харків. : Компанія СМІТ, 2005. – 319 с.
8. Алгоритмізація та програмування. Програмування мовою високого рівня С++ : метод. вказівки до викон. лаб. робіт для напряму підгот. 6.050101 «Комп'ютерні науки» » / Уклад. С. М. Алхімова. – К.: НТУУ «КПІ», 2015. – 108 с.
9. Козак, Л. І. Основи програмування : навч. посібник / Л. І. Козак. - Львів : Новий світ-2000,

2018. – 328 с.
10. Дудзяний І. М. Програмування мовою С++. Частина 1 : Парадигма процедурного програмування : навчальний посібник. Львів : ЛНУ імені Івана Франка, 2013. – 468 с.
  11. Ковалюк, Т. В. Основи програмування. – К: Видавнича група ВНУ, 2005. – 384 с.
  12. Ковалюк, Т. В. Алгоритмізація та програмування: підручник. – Львів: Магнолія 2006, 2013. – 400 с.
  13. Матвієнко, М. П. Математична логіка та теорія алгоритмів: навч. посібник / Матвієнко М. П. Шаповалов С. П. –К.: Ліра, 2015. – 212 с.
  14. Васильєв, О.М. Програмування на С++ в прикладах і задачах : навч. посібник / О. М. Васильєв. – Київ : Ліра-К, 2017. – 382 с.
  15. Белов, Т.О. Вступ до програмування мовою С++. Організація обчислень: навч. посіб. / Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К.: Видавничо-поліграфічний центр «Київський університет», 2012. – 175 с.
  16. Азарян, А.А. Основи алгоритмізації та програмування : навч. посіб. / А. А. Азарян, Н. О. Карабут, Т.П. Козикова, О. Г. Рибальченко, А. А. Трачук, Н. Н. Шаповалова. – Кривий Ріг: Вид-во Оксандринт, 2014. – 308 с.
  17. Булига, О. А. Конспекти лекцій з алгоритмізації та програмування. Мова С++ : навч. посіб. / Булига Олена Анатоліївна. – Київ ; Харків : Панов А. М. [вид.], 2022. – 142 с.
  18. Булига, О. А. Лабораторний практикум з алгоритмізації та програмування. Мова С++ : навч. посіб. / Булига Олена Анатоліївна. - Київ ; Харків : Панов А. М. [вид.], 2022. – 111 с.
  19. Щедрина, О. І. Алгоритмізація та програмування процедур обробки інформації С++ : навч. посібник / О.І. Щедрина. – К. : КНЕУ, 2001. – 240 с.
  20. Програмування : навч. посіб. для студ. вищ. навч. закл. / В. В. Зубенко, Л. Л. Омельчук ; Київ. нац. ун-т ім. Тараса Шевченка. - К. : Київський університет, 2011. - 623 с.

## Інформаційні ресурси

1. Інтернет-портал організаційно-методичного забезпечення дистанційних олімпіад з програмування для обдарованої молоді навчальних закладів України : [Електронний ресурс]. – Режим доступу : <http://www.e-olimp.com.ua/ua>
2. International Standard ISO/IEC 14882:2014(E) – Programming Language C++ : [Електронний ресурс]. – Режим доступу : <https://isocpp.org/std/the-standard>
3. MS Visual Studio : [Електронний ресурс]. – Режим доступу: <http://visualstudio.com>
4. MSDN : [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com>

## Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компонента) і самостійна робота студента

## Лекції

### Лекція 1. Вступна

*Питання, що розглядаються:*

- 1) мета дисципліни «Алгоритмізація та програмування»;
- 2) алгоритм і програма: визначення і сутність понять;
- 3) алгоритмічні мови програмування.

*Література:*

основна – [1, 2];  
додаткова – [1- 4].

*Завдання на СРС:*

вивчити конспект лекції; ознайомитися із списком рекомендованої літератури; знати визначення і сутність основних понять; ознайомитися із вимогами до виконання комп'ютерних, структурою звіту з комп'ютерного практикуму.

### Лекція 2. Основи алгоритмізації

*Питання, що розглядаються:*

- 1) етапи розв'язування обчислювальних задач на комп'ютері;
- 2) алгоритм, його властивості;
- 3) способом опису алгоритмів, блок-схеми;
- 4) приклади базових алгоритмічних структур.

*Література:*

основна – [1, 2];

додаткова – [1- 4].

*Завдання на СРС:*

вивчити конспект лекції; вивчити етапи розв'язування обчислювальних задач на комп'ютері; знати поняття алгоритму, його властивості; знати найбільш поширені способи опису алгоритмів: словесний запис, графічні схеми алгоритмів (блок-схеми), псевдокод (формальні алгоритмічні мови); вивчити графічні елементи блок-схем та відповідні до кожного з елементів функції; розглянути базові алгоритмічні структури.

### **Лекція 3. Позиційні системи числення**

*Питання, що розглядаються:*

- 1) основні поняття систем числення;
- 2) правила переведення чисел з однієї системи числення в іншу;
- 3) арифметичні операції.

*Література:*

основна – [2, 3];  
додаткова – [1- 4].

*Завдання на СРС:*

вивчити конспект лекції; знати основні поняття систем числення; знати особливості роботи з позиційними системами числення; знати правила переведення чисел з однієї системи числення в іншу; знати як переводити числа з десяткової у двійкову, з двійкової у десяткову, з десяткової у вісімкову, з вісімкової у десяткову; з десяткової у шістнадцяткову, з шістнадцяткової у десяткову, з з вісімкової у двійкову, з двійкової у вісімкову, з шістнадцяткової у двійкову, з двійкової у шістнадцяткову систем числення; знати особливості виконання арифметичних операцій (додавання, віднімання, множення, ділення) над числами у різних системах числення.

### **Лекція 4. Програмування та алгоритмічні мови**

*Питання, що розглядаються:*

- 1) програмування та алгоритмічні мови, основні поняття;
- 2) трансляція коду програми;
- 3) компілятори і інтерпретатори.

*Література:*

основна – [3 (С. 6-12, 467-476)];  
додаткова – [5-8, 10].

*Завдання на СРС:*

вивчити конспект лекції; знати основні поняття за темою лекції; знати, що може бути отримане у результаті трансляції, особливості та характеристики машинних інструкцій, мікрокоду, байткоду, об'єктного модуля, програмного коду; знати класифікацію трансляторів залежно від того, що подають на вхід і що отримують в результаті процесу трансляції; знати принципову різницю між компіляторами і інтерпретаторами.

### **Лекція 5. Вступ до мови програмування високого рівня С++**

*Питання, що розглядаються:*

- 1) історія розвитку і стандарт мови С++;
- 2) лексеми мови С++;
- 3) структура програм, що написані мовою С++;
- 4) базові функції стандартної бібліотеки.

*Література:*

основна – [3 (С. 6-12, 467-476)];  
додаткова – [2, 4-6, 10].

*Завдання на СРС:*

вивчити конспект лекції; ознайомитися із історією розвитку і стандартом мови С++; знати алфавіт і лексеми мови С++ (ідентифікатори, оператори, константи, ключові слова, роздільники); знати структура програм, що написані мовою С++, вміти пояснювати на прикладі.

### **Лекція 6. Інтегроване середовище розробки програм Microsoft Visual Studio**

*Питання, що розглядаються:*

- 1) основні стадії створення виконуваного файлу;
- 2) поняття інтегрованого середовища розробки;
- 3) створення проекту у Microsoft Visual Studio;
- 4) компіляція і запуск програмного застосунку.

*Література:*

основна – [3 (С. 6-12, 467-476)];  
додаткова – [1- 4].

*Завдання на СРС:*

вивчити конспект лекції; знати основні стадії створення виконуваного файлу; знати, що таке процеси компіляції та компонування; знати, що таке інтегроване середовище розробки; знати, в яких варіантах доступна Microsoft Visual Studio, їх різницю; знати специфіку створення проекту у Microsoft Visual Studio; знати механізми редагування, компіляції та запуску простої програми, написаної на С++ з використанням Microsoft Visual Studio.

## Лекція 7. Концепція типів даних. Об'явлення, визначення та ініціалізація змінних

*Питання, що розглядаються:*

- 1) зберігання інформації у пам'яті комп'ютера;
- 2) поняття типу даних;
- 3) особливості стандартних типів мови C++;
- 4) внутрішні формати роботи з даними в пам'яті комп'ютера;
- 5) об'явлення, визначення та ініціалізація змінних.

*Література:*

основна – [3, 4];

додаткова – [2, 4-6, 10].

*Завдання на СРС:*

вивчити конспект лекції; знати, в який чин інформація зберігається у пам'яті комп'ютера; знати поняття типу даних і особливості стандартних типів мови програмування C++; знати розміри, що займають змінні основних типів даних мови C++; знати діапазони значень, які можуть набувати змінні основних типів даних мови C++; знати внутрішні формати роботи з даними в пам'яті комп'ютера (запис чисел з фіксованою крапкою і запис чисел з рухомою крапкою); знати, що таке машинний нуль і машинна нескінченність; знати поняття об'явлення, визначення та ініціалізації змінних; вміти пояснювати основні операції та вирази мови C++, введення / виведення даних засобами форматowanego та потокового введення-виведення, прості команди C++, механізм привласнення значення і виразу, булеві вирази й оператори.

## Лекція 8. Алгоритми розгалуженої структури

*Питання, що розглядаються:*

- 1) умовний оператор if;
- 2) умовний вираз;
- 3) оператор вибору варіантів switch.

*Література:*

основна – [1 (С. 46-68), 3, 4];

додаткова – [2, 4-6, 10].

*Завдання на СРС:*

вивчити конспект лекції; вивчити форму та правила використання умовного оператора if, оператора перемикання switch; вивчити структури блок-схем, які використовуються для опису розгалуженого обчислювального процесу; проаналізувати випадки використання оператора безадресного переходу break для виходу з оператора switch.

## Лекція 9. Алгоритми циклічної структури

*Питання, що розглядаються:*

- 1) оператори циклу for, while, do-while;
- 2) оператори безумовної адресації break та continue.

*Література:*

основна – [1 (С. 46-68), 3, 4];

додаткова – [2, 4-6, 10].

*Завдання на СРС:*

вивчити конспект лекції; вивчити форму та правила використання операторів циклів for, while, do-while; вивчити структури блок-схем, які використовуються для опису циклічного обчислювального процесу; проаналізувати випадки використання оператора безадресного переходу break для виходу з операторів циклів for, while та do-while; проаналізувати випадки використання оператора continue для операторів циклів for, while та do-while.

## Лекція 10. Показчики та посилання

*Питання, що розглядаються:*

- 1) поняття показчика, робота з ним;
- 2) оператор отримання адреси змінної;
- 3) оператор розіменування показчика;
- 4) операції з показчиками;
- 5) принципом асоціативності під час виконання операцій над показчиком;
- 6) особливості вживання модифікатора const з показчиками;
- 7) поняття посилання, робота з ним.

*Література:*

основна – [1 (С. 74-88), 3];

додаткова – [1- 4].

*Завдання на СРС:*

вивчити конспект лекції; проаналізувати особливості використання символу зірочки " \* " у випадках для визначення вказівників та як операції розіменування; проаналізувати особливості використання символу амперсанда "&" у випадках для визначення посилання та як операції взяття адреси; визначити послідовність виконання операцій з показчиком, який має префіксу та постфіксу операції; проаналізувати особливості вживання модифікатора const з показчиками на конкретних прикладах; визначити основні приклади некоректного використання показчиків.

## Лекція 11. Розробка функцій користувача та робота з ними

*Питання, що розглядаються:*

- 1) оголошення (прототип), визначання та виклик функцій користувача;
- 2) формальні та фактичні параметри функції;
- 3) способи передачі параметрів у функцію;
- 4) константні параметри функції;
- 5) функція з параметрами за замовчуванням;
- 6) покажчики на функції;

*Література:*

основна – [1 (С. 152-161), 3];  
додаткова – [2, 4-6, 10, 14].

*Завдання на СРС:*

вивчити конспект лекції; два способи визначення функцій користувача в програмі; визначити особливості роботи з прототипами функцій користувача; проаналізувати різницю між формальними та фактичними параметрами функцій; визначити правила, за допомогою яких підтримується відповідність між формальними та фактичними параметрами; проаналізувати механізми передачі параметрів у функції за значенням та за адресою; визначити призначення та особливості використання покажчика на будь-який тип даних при роботі з функціями; проаналізувати механізм роботи з покажчиками на функції.

## Лекція 12. Рекурсія

*Питання, що розглядаються:*

- 1) рекурсивний виклик функції, глибина рекурсії;
- 2) пряма і непряма рекурсія;
- 3) обмеження на кількість рекурсивних викликів;
- 4) рекурсія та цикли: схожість та відмінності;
- 5) ефективність використання рекурсії.

*Література:*

основна – [1 (С. 152-161), 3];  
додаткова – [9, 11-15].

*Завдання на СРС:*

вивчити конспект лекції; особливості використання рекурсивних функцій в мові програмування C++; знати, що може призводити до рекурсивного зациклювання програми; розуміти, в яких випадках використання рекурсії не є ефективним.

## Лекція 13. Область видимості та час існування змінних

*Питання, що розглядаються:*

- 1) область видимості (дії) змінних, локальні та глобальні змінні;
- 2) простір імен;
- 3) час існування змінних;
- 4) поняття статичних локальних змінних.

*Література:*

основна – [1 (С. 69-74), 3];  
додаткова – [2, 4-6, 10, 16-18].

*Завдання на СРС:*

вивчити конспект лекції; визначити особливості роботи з глобальними (зовнішніми) та локальними (внутрішніми) змінними; визначити, що саме потребує розміщення в пам'яті при виконанні програми на мові C++; використання пам'яті стека та купи; автоматичні та статичні змінні в програмах на мові C++; вивчити основні правила роботи з просторами імен.

## Лекція 14. Системи контролю версій коду

*Питання, що розглядаються:*

- 1) системи контролю версій коду, їх призначення;
- 2) основні поняття: trunk, branch, check-out, check-in (code commit);
- 3) виникнення конфліктів, їх вирішення.

*Література:*

основна – [1 (С. 69-74), 3];  
додаткова – [4-6, 10].

*Завдання на СРС:*

вивчити конспект лекції; визначити особливості роботи із системами контролю версій коду; знати як вирішувати конфлікти у ручному режимі, принцип вирішення конфліктів у автоматичному режимі.

## Комп'ютерні практикуми

**Основна ціль комп'ютерних практикумів** полягає в оволодінні студентами мовою програмування високого рівня C++; вивченні та оволодінні основних засад програмування типових задач мовою C++; отриманні базових навичок роботи в інтегрованому середовищі розробки програмного забезпечення Microsoft Visual Studio.

### Комп'ютерний практикум 1. Блок-схеми алгоритмів виконання програм

*Питання, що розглядаються:*

- 1) правила побудови блок-схем;
- 2) правила визначення блок-схем алгоритмів для написання програм на мові програмування C++;
- 3) графічні елементи та базові структурами блок-схем.

*Література:*

основна – [5];  
додаткова – [1- 4, 18-20].

*Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке алгоритм; які існують способи опису алгоритмів; використовуючи словесний запис опису алгоритмів, скласти алгоритм переходу вулиці на світлофор; перелічити основні графічні елементи блок-схем; що визначають лінії, які з'єднують графічні блоки на блок-схемах алгоритмів; визначити призначення графічного блоку «Визначений процес»; чи може в блок-схемі, в якій використано «Логічний блок», бути два графічних блоки «Кінець»; навести приклад використання графічного блоку «Перехід» для з'єднання блок-схеми між двома сторінками; правила оформлення блок-схем.

## **Комп'ютерний практикум 2. Позиційні системи числення**

*Питання, що розглядаються:*

- 1) позиційні системи числення;
- 2) двійкова система числення;
- 3) вісімкова система числення;
- 4) шістнадцяткова система числення;
- 5) переведення з десяткової системи числення в інші позиційні системи числення;
- 6) арифметичні операції в позиційних системах числення.

*Література:*

основна – [2];  
додаткова – [1, 2, 18-20].

*Завдання на СРС:*

вивчити особливості роботи з позиційними системами числення; ознайомитися з переводом з десяткової в вісімкову, з вісімкової в десяткову; з десяткової в шістнадцяткову та з шістнадцяткової в десяткову систем числення; визначити особливості роботи з двійковою, вісімковою та шістнадцятковою системами числення при розробці програм мовою C++; вивчити особливості арифметичних операцій в позиційних системах числення.

## **Комп'ютерний практикум 3. Інтегроване середовище розробки Microsoft Visual Studio**

*Питання, що розглядаються:*

- 1) ознайомитися з основами роботи в середовищі розробки Microsoft Visual Studio;
- 2) проекти для розробки консольних програмних застосунків;
- 3) файли для розробки програмного коду всередині проектів консольних програм;
- 4) основні властивості консольних проектів, їх призначення;
- 5) призначення всіх файлів консольних проектів.

*Література:*

основна – [5];  
додаткова – [1- 4, 18-20].

*Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття; знати що таке процес компіляції програми та що таке процес компанування програми; ознайомитися з основними головними рисами комерційних варіантів Microsoft Visual Studio; вміти виконувати створення нового проектного файлу в середовищі Microsoft Visual Studio та додавання нового файлу вихідного коду в середовищі Microsoft Visual Studio.

## **Комп'ютерний практикум 4. Внутрішні формати роботи з даними в пам'яті комп'ютера**

*Питання, що розглядаються:*

- 1) подання цілих чисел в комп'ютері;
- 2) арифметичні дії з цілими числами в комп'ютері;
- 3) подання дійсних чисел в комп'ютері;
- 4) арифметичні дії над нормалізованими числами в комп'ютері.

*Література:*

основна – [5];  
додаткова – [1, 2, 18-20].

*Завдання на СРС:*

ознайомитися з поданням цілих та дійсних чисел в комп'ютері; визначити особливості проведення арифметичних дій з цілими та нормалізованими числами в комп'ютері.

## **Комп'ютерний практикум 5. Алгоритми послідовної (лінійної) структури**

*Питання, що розглядаються:*

- 1) особливості використання типів даних (char, int, float, double);
- 2) операції мови C++;
- 3) функції введення-виведення;
- 4) стандартні математичні функції;
- 5) програмування послідовних (лінійних) обчислювальних процесів.



### *Література:*

основна – [3-5];  
додаткова – [1-4, 18-20].

### *Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке алгоритм послідовної (лінійної) структури, програма послідовної (лінійної) структури; дати визначення оператора, операнда та операції; перелічити основні стандартні типи даних визначені в мові С++; що таке операції декремента, інкремента; як визначаються коментарі у програмному коді, навести відповідні приклади; яка різниця під час використання форматowanego та потокового введення-виведення даних; що визначає ширина поля вводу/виводу (навести відповідні приклади); як визначити максимальну кількість цифр після десяткової точки для виведення на екран даних дійсного типу; для чого необхідні стандартні файли заголовків.

## **Комп'ютерний практикум 6. Логічні основи комп'ютерів**

### *Питання, що розглядаються:*

- 1) алгебра логіки;
- 2) логічний вислів;
- 3) логічна формула;
- 4) алгебра логіки та двійкове кодування в комп'ютерах;
- 5) логічний елемент комп'ютера;
- 6) схема «І», «АБО», «НІ», «І-НІ», «АБО-НІ»;
- 7) таблиця істинності;
- 8) спрощення логічних формул;
- 9) вирішення логічних задач.

### *Література:*

основна – [5];  
додаткова – [1, 2, 18-20].

### *Завдання на СРС:*

ознайомитися з основами алгебри логіки, вивчити особливості застосування алгебри логіки з двійковим кодуванням в комп'ютерах; визначити особливості роботи логічних елементів комп'ютера, схеми «І», «АБО», «НІ», «І-НІ», «АБО-НІ»; ознайомитися з поняттям таблиць істинності та навчитися працювати з ними; навчитися спрощувати логічні формули та вирішувати логічні задачі.

## **Комп'ютерний практикум 7. Алгоритми розгалуженої структури (умовний оператор)**

### *Питання, що розглядаються:*

- 1) можливості створення розгалужених структур;
- 2) формат умовного оператора if;
- 3) програмування розгалужених обчислювальних процесів на основі використання умовного оператора.

### *Література:*

основна – [3-5];  
додаткова – [1- 4, 18-20].

### *Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке алгоритм розгалуженої структури; синтаксис умовного оператора; яка різниця між умовними операторами з однією та двома гілками; чи можливе існування тільки гілки else (без операторів if умови) в умовному операторі; коли в умовному операторі слід використовувати фігурні дужки «{» та «}», а коли ні; що таке логічний вираз; логічні операції, що використовуються в логічних виразах при написанні програм мовою С++; яка операція («&&» чи «||») має більший пріоритет, навести відповідний приклад для пояснення відповіді; як будується складна умова, що об'єднує декілька умов.

## **Комп'ютерний практикум 8. Алгоритми розгалуженої структури (оператор перемикання)**

### *Питання, що розглядаються:*

- 1) особливості використання оператора перемикання та визначення ситуацій, що потребують його використання;
- 2) формат оператора перемикання switch, основні його відмінності від умовного оператора для написання програм з використанням алгоритмів розгалуженої структури;
- 3) програмування розгалужених обчислювальних процесів на основі використання оператора перемикання.

### *Література:*

основна – [3-5];  
додаткова – [1- 4, 18-20].

### *Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: в яких випадках слід використовувати оператор перемикання switch; синтаксис оператора перемикання; чи можуть case-гілки з одного оператора switch мати однакові значення міток; як default впливає на виконання тіла оператора перемикання; чи можуть гілки case та default бути розміщені в довільному порядку; чи можуть бути умовні вирази варіантів одного оператора switch однаковими; що може бути використано в якості міток в операторі перемикання switch; навіщо в операторі switch використовувати оператор переривання break; коли краще використовувати умовний оператор, а коли

оператор перемикання; чи може оператор перемикання switch мати вкладений оператор умови if, ще один вкладений оператор перемикання switch, відповідь пояснити.

### **Комп'ютерний практикум 9. Розгалуження в циклі**

*Питання, що розглядаються:*

- 1) розгалуження в циклі мовою C++;
- 2) ситуації, що потребують використання розгалуження в циклі;
- 3) формат оператора циклів for;
- 4) програмування розгалуження всередині циклічного обчислювального процесу.

*Література:*

основна – [3-5];  
додаткова – [1- 4, 18-20].

*Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: для яких цілей використовують цикли в програмуванні; що таке тіло циклу; у яких випадках краще використовувати циклу з параметром; зазначити порядок передачі керування під час виконання циклу for; який тип може мати лічильник циклу for; якщо початковій значення лічильника оператора циклу for виявиться менше кінцевого значення, чи буде виконуватися тіло циклу хоча б один раз; чи можна змінити значення лічильника циклу з параметром for в середині тіла цього циклу; скільки разів буде виконане повторення тіла оператора циклу for, якщо параметр циклу приймає всі цілі значення від 1 до 5, всі парні значення від 2 до 21, всі значення від 10 до 50 з кроком 7, всі значення від 2,5 до 8,3 з кроком 0,4; як виконати пропуск ітерації оператора циклу for; як у операторі циклу for організувати одразу кілька змінних-лічильників.

### **Комп'ютерний практикум 10. Алгоритми циклічної структури**

*Питання, що розглядаються:*

- 1) циклічні структури мови C++;
- 2) ситуації, що потребують використання циклів;
- 3) формат операторів циклів for, while, do-while;
- 4) основні схожості та відмінності операторів циклів для написання програм з використанням алгоритмів циклічної структури;
- 5) програмування циклічних обчислювальних процесів.

*Література:*

основна – [3-5];  
додаткова – [1- 4, 18-20].

*Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: для яких цілей використовують цикли в програмуванні; чи можна змінити значення лічильника циклу з параметром for в середині тіла цього циклу; який тип може мати лічильник циклу for; чи може тіло циклу while бути порожнім; як організувати нескінчений цикл з використанням оператора for, оператора while, оператора do-while; в чому полягають схожість та різниця між циклами з передумовою та з післяумовою; для чого потрібен оператор безадресного переходу break; для чого потрібен оператор безадресного переходу continue; що таке вкладені цикли; як можна вийти із тіла вкладеного циклу в тіло основної програми.

### **Комп'ютерний практикум 11. Функції користувача**

*Питання, що розглядаються:*

- 1) функції користувача;
- 2) правила роботи з функціями користувача;
- 3) програмування функцій користувача для виконання бітових операцій над цілими числами.

*Література:*

основна – [3-5];  
додаткова – [1- 4, 18-20].

*Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке прототип (сигнатура) функції, навіщо він потрібен; чим відрізняється визначення функції від об'явлення її прототипу; чи може в одній функції зустрічатися декілька операторів return; скільки значень може повернути функція; як слід організувати функції, якщо необхідно повернути значення декількох змінних у точку виклику цієї функції; що таке формальні параметри функції; що таке фактичні параметри функції; які особливості передачі параметрів у функції в мові C++; які є способи виклику функції; значення якого типу повертає функція, якщо тип не вказано.

### **Комп'ютерний практикум 12. Рекурсія**

*Питання, що розглядаються:*

- 1) властивості рекурсивності при розробці функцій за допомогою мови програмування C++;
- 2) застосування глобальних та локальних змінних разом з рекурсивними функціями;
- 3) механізми передачі параметрів до функцій;
- 4) програмування рекурсивних функцій.

*Література:*

основна – [3-5];  
додаткова – [1- 4, 18-20].

#### *Завдання на СРС:*

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповідати на питання: дати визначення рекурсії та наведіть приклади її використання в різних галузях знань; в яких задачах доцільно використовувати рекурсивні функції; що спільного та в чому полягає різниця між циклічними та рекурсивними способами організації розрахунків; визначити основні особливості роботи рекурсивних функцій в мові програмування C++; що таке пряма та непряма (опосередкована) рекурсії; коли виконується завершення рекурсивних викликів, що таке рекурсивне зациклювання та до чого воно призводить; що таке область дії змінних; як необхідно виконати об'яву змінної, щоб доступ до неї був можливим з будь-якої функції поточного файлу програми; як змінюються значення локальних та глобальних змінних при використанні рекурсивних функцій; визначте головне обмеження при використанні рекурсії.

### **Комп'ютерний практикум 13. Проведення МКР**

*Питання, що виносяться на МКР пов'язані із засвоєнням студентами наступних тем:*

**Тема 1.** Основи алгоритмізації

**Тема 2.** Позиційні системи числення

**Тема 3.** Елементи алгоритмічних мов: поняття типу даних, ідентифікатори, змінні, константи, операції

**Тема 4.** Структурне програмування: послідовність, розгалуження та цикли

**Тема 5.** Процедурно-орієнтоване програмування. Рекурсія

*Література:*

основна – [1-5];

додаткова – [1-20].

*Завдання на СРС:*

вивчити конспекти лекцій, теоретичні відомості щодо організації програм мовою високого рівня C++.

## **Політика та контроль**

### **6. Політика навчальної дисципліни (освітнього компонента)**

Форми організації освітнього процесу, види навчальних занять і оцінювання результатів навчання регламентуються Положенням про організацію освітнього процесу в Національному технічному університеті України «Київському політехнічному інституті імені Ігоря Сікорського».

**Правила відвідування занять.** Відвідування є обов'язковим (за винятком випадків, коли існує поважна причина, наприклад, хвороба чи дозвіл працівників деканату). Якщо студент не може бути присутнім на заняттях, він все одно несе відповідальність за виконання завдань, що проводились в комп'ютерному класі.

**Правила поведінки на заняттях.** Під час виконання комп'ютерних практикумів студент може користуватися ноутбуком, мобільним телефоном або іншими пристроями для пошуку інформації на гугл-диску викладача чи в інтернеті. Проте під час лекційних занять та обговорення завдань комп'ютерних практикумів зазначеними раніше пристроями користуватися неможна. Це відволікає викладача і студентів групи та перешкоджає навчальному процесу. Якщо мається намір використовувати ноутбук або інший пристрій для аудіо- чи відеозапису, необхідно заздалегідь отримати дозвіл викладача. Під час лекційних занять заборонено відволікати викладача, усі питання, уточнення та ін. студенти задають в кінці лекції або відведений для цього час.

### **Виконання завдань контрольних заходів**

Плагіат та інші форми нечесної роботи неприпустимі. Всі комп'ютерні студенти мають виконувати самостійно із використанням рекомендованої літератури й отриманих знань та навичок. Цитування в письмових роботах допускається тільки із відповідним посиланням на авторський текст. Недопустимі підказки і списування у ході захисту комп'ютерних практикумів (лабораторних робіт), на модульній контрольній роботі, на екзамені.

**Комп'ютерні практикуми** захищаються особисто з попередньою перевіркою теоретичних знань, які необхідні для виконання завдань за темою комп'ютерного практикуму. Перевірка практичних результатів включає перевірку коду та виконання програмного застосунку.

Модульна контрольна робота проводиться письмово без застосування допоміжних засобів (мобільні телефони, планшети, література та ін.) за принципом хронометражу часу виконання: доступ до контрольного завдання МКР відкривається викладачем у заздалегідь оголошений

момент на визначений період часу. Результати модульної контрольної роботи оголошуються студентам на наступному занятті.

**Порядок зарахування пропущених занять.** Відпрацювання пропущеного заняття з лекційного курсу здійснюється шляхом підготовки і захисту реферату за відповідною темою. Захист реферату відбувається відповідно до графіку консультацій викладача, з яким можна ознайомитись на кафедрі. Відпрацювання пропущеної комп'ютерного практикуму здійснюється шляхом самостійного виконання завдання і його захисту відповідно до графіку консультацій викладача.

**Політика щодо академічної доброчесності.** 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>.

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

Рейтинг студента за освітній компонент складається з балів, що студент отримує за:

- виконання та захист комп'ютерних практикумів із 12 тем:
  1. Блок-схеми алгоритмів виконання програм (Тема 1. Основи алгоритмізації);
  2. Позиційні системи числення (Тема 2. Позиційні системи числення);
  3. Інтегроване середовище розробки програм Microsoft Visual Studio (Тема 3. Елементи алгоритмічних мов: поняття типу даних, ідентифікатори, змінні, константи, операції);
  4. Внутрішні формати роботи з даними в пам'яті комп'ютера (Тема 3. Елементи алгоритмічних мов: поняття типу даних, ідентифікатори, змінні, константи, операції);
  5. Алгоритми послідовної (лінійної) структури (Тема 4. Структурне програмування: послідовність, розгалуження та цикли);
  6. Логічні основи комп'ютерів (Тема 4. Структурне програмування: послідовність, розгалуження та цикли);
  7. Алгоритми розгалуженої структури (умовний оператор). (Тема 4. Структурне програмування: послідовність, розгалуження та цикли);
  8. Алгоритми розгалуженої структури (оператор перемикання). (Тема 4. Структурне програмування: послідовність, розгалуження та цикли);
  9. Розгалуження в циклі (Тема 4. Структурне програмування: послідовність, розгалуження та цикли);
  10. Алгоритми циклічної структури (Тема 4. Структурне програмування: послідовність, розгалуження та цикли);
  11. Функції користувача (Тема 5. Процедурно-орієнтоване програмування. Рекурсія);
  12. Рекурсія (Тема 5. Процедурно-орієнтоване програмування. Рекурсія)
- одну модульну контрольну роботу.

Максимальна сума балів за всі контрольні заходи протягом семестру складає:

$$12 \text{ комп'ютерних практикумів} \cdot 7 \text{ балів} + 4 \text{ питання МКР} \cdot 4 \text{ бали} = 100 \text{ балів.}$$

### Поточний контроль

#### Комп'ютерні практикуми

Ваговий бал – 7.

Максимальна кількість балів за всі роботи дорівнює  $7 \text{ балів} \times 12 = 84 \text{ балів}$ .

*Критерії оцінювання:*

- 7 балів** – вірно виконані реалізація алгоритмів, розрахункова частина роботи та досконало проведений аналіз отриманих результатів, блок-схеми без зауважень;
- 5.6...6.9 балів** – вірно виконані реалізація алгоритмів та розрахункова робота, але з недостатньо проведеним аналізом отриманих результатів та теоретичним обґрунтуванням, помилки в записі блок-схем;
- 4.2...5.5 балів** – робота виконана з грубими помилками;
- 0 балів** – невірно виконано.

#### Модульна контрольна робота

Ваговий бал – 4.

Максимальна кількість балів за контрольну роботу дорівнює  $4 \text{ бали} \times 4 \text{ питання} \times 1 \text{ МКР} = 16 \text{ балів}$

*Критерії оцінювання:*

- 2.4 ... 4 бали** – повна вірна відповідь на питання (відповідь містить не менше ніж 60 % необхідної інформації);
- 0 балів** – відсутність відповіді або невірна відповідь на питання (менше 60% необхідної інформації).

### Календарний контроль

Проміжна атестація студентів (далі – атестація) є календарним контролем. Метою проведення атестації є підвищення якості навчання студентів та моніторинг виконання графіка освітнього процесу студентами.

Умовою першої атестації є отримання не менше 27 балів та захист всіх **комп'ютерних практикумів** (на час атестації). Умовою другої атестації є отримання не менше 45 балів та захист всіх **комп'ютерних практикумів** (на час атестації).

### **Семестровий контроль**

Умовами допуску до заліку є:

- зарахування всіх **комп'ютерних практикумів**;
- написана МКР.

Студенти, які виконали всі умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань.

Студенти, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, виконують залікову контрольну роботу.

Студенти, які у семестрі отримали більше 60 балів, можуть взяти участь у заліковій контрольній роботі. У цьому разі бали, отримані ними на заліковій контрольній роботі, є остаточними.

Залікова контрольна робота оцінюється зі 100 балів. Кількість запитань у кожному білеті – 3. Ваговий бал першого запитання – 25, другого – 25, третього – 50.

*Критерії оцінювання першого та другого запитань на екзамені:*

- 23...25 балів** – змістовна відповідь на теоретичне питання білету;
- 17...22 балів** – добра відповідь на питання, але з невеликими зауваженнями;
- 15...16 балів** – задовільна відповідь на питання (є декілька грубих помилок у відповіді);
- 0 бали** – більше двох грубих помилок/незнання питання

*Критерії оцінювання третього запитання на екзамені:*

- 45...50 балів** – правильна програмно-алгоритмічна реалізація розв'язку задачі білету;
- 35...44 балів** – допущено несуттєві неточності в програмній реалізації;
- 30...34 балів** – помилки в рахунках, не викладено основні етапи алгоритмічної частини;
- 0 балів** – незнання розв'язку задачі, грубі помилки.

### **Таблиця відповідності рейтингових балів оцінкам за університетською шкалою**

Сума балів, отриманих студентом, переводиться до оцінки згідно з таблицею:

<b>Бали</b>	<b>Оцінка</b>
100...95	Відмінно
94...85	Дуже добре
84...75	Добре
74...65	Задовільно
64...60	Достатньо
Менше 60	Незадовільно
Є не зараховані лабораторні роботи або не написана МКР	Не допущено

### **8. Додаткова інформація з дисципліни (освітнього компонента)**

**Дистанційне навчання.** В умовах дистанційного режиму організація освітнього процесу здійснюється з використанням технологій дистанційного навчання: платформи дистанційного навчання «Сікорський» та «Електронний кампус». Навчальний процес у дистанційному режимі здійснюється відповідно до затвердженого розкладу навчальних занять. Заняття проходять з використанням сучасних ресурсів проведення онлайн-зустрічей (організація відео-конференцій).

### **Робочу програму навчальної дисципліни (силабус):**

Складено доцентом кафедри БМК, к.т.н., Алхімовою Світланою Миколаївною

Ухвалено кафедрою біомедичної кібернетики (протокол №1 від 31 серпня 2023 року)

Погоджено Методичною комісією факультету (протокол №1 від 1 вересня 2023 року)