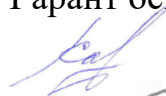


Міністерство освіти і науки України  
Національний аерокосмічний університет ім. М.Є. Жуковського  
“Харківський авіаційний інститут”

Кафедра мехатроніки та електротехніки (№ 305)

**ЗАТВЕРДЖУЮ**

Гарант освітньої програми

  
(підпис)

Кочук С.Б.  
(ініціали та прізвище)

«30» серпня 2021 р.

## **РОБОЧА ПРОГРАМА ОБОВ’ЯЗКОВОЇ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

### **Програмування та алгоритмічні мови**

**Галузь знань:** 15 «Автоматизація та приладобудування»

**Спеціальність:** 151 «Автоматизація та комп’ютерно-інтегровані технології»

**Освітня програма:** «Комп’ютерно-інтегровані технологічні процеси і виробництва»

**Форма навчання:** **денна**

**Рівень вищої освіти:** перший (бакалаврський)

**Харків 2021 рік**

Розробник: Лутай Л. М., доцент каф. №305, к.т.н., доцент

  
(підпис)

Робочу програму розглянуто на засіданні кафедри мехатроніки та електротехніки

Протокол № 1 від « 30 » серпня 2021 р.

В. о. завідувача кафедри  
к.т.н., доцент



(К. Ф. Фомичов)

## Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 4	<p style="text-align: center;"><b>Галузь знань</b> <u>15 «Автоматизація та приладобудування»</u></p> <p style="text-align: center;"><b>Спеціальність</b> <u>151 «Автоматизація та комп'ютерно-інтегровані технології»</u></p> <p style="text-align: center;"><b>Освітня програма</b> <u>«Комп'ютерно-інтегровані технологічні процеси і виробництва»</u></p> <p style="text-align: center;"><b>Рівень вищої освіти:</b> <u>перший (бакалаврський)</u></p>	Обов'язкова
Кількість модулів – 2		<b>Навчальний рік</b>
Кількість змістових модулів – 2		2021/ 2022
Індивідуальне завдання – РР		<b>Семестр</b>
		3-й
Загальна кількість годин – 56/120		<b>Лекції</b>
		16 годин
Кількість тижневих годин для денної форми навчання: аудиторних – 3,5 самостійної роботи студента – 4		<b>Практичні, семінарські</b>
		16 годин
		<b>Лабораторні</b>
	24 години	
	<b>Самостійна робота</b>	
	64 годин	
	<b>Вид контролю</b>	
	модульний контроль, залік	

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: 56/64.

\*Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

## 2. Мета та завдання навчальної дисципліни

**Мета:** отримати знання та практичні навички створення програм для роботи автоматизованих виробничих та мехатронних систем.

**Завдання:** вивчення мов HTML, PHP, C++ та системами управління базами даних MySQL.

### **Компетентності, які набуваються:**

1. Здатність вчитися й оволодівати сучасними знаннями.
2. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
3. Здатність застосовувати знання у практичних ситуаціях.
4. Навички використання інформаційних і комунікаційних технологій.
5. Здатність працювати в команді.
6. Здатність застосовувати відповідні математичні, наукові і технічні методи, а також комп'ютерне програмне забезпечення для аналізу і синтезу систем автоматизації.
7. Здатність застосовувати методи системного аналізу, математичного моделювання, ідентифікації та числові методи для розроблення математичних моделей окремих елементів та систем автоматизації в цілому, для аналізу якості їх функціонування із використанням новітніх комп'ютерних технологій.
8. Здатність обґрунтовувати вибір технічних засобів автоматизації на основі розуміння принципів їх роботи, аналізу їх властивостей, призначення і технічних характеристик; налагоджувати та експлуатувати системи автоматизації.
9. Здатність використовувати для вирішення професійних завдань новітні технології у галузі автоматизації та комп'ютерно-інтегрованих технологій.
10. Здатність обґрунтовувати вибір та розробляти прикладне програмне забезпечення для мікропроцесорних систем керування на базі локальних засобів автоматизації, промислових логічних контролерів.
11. Здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями для вирішення задач автоматизації, зокрема розробки 3-D моделей, електричних схем та планування рішень.

### **Очікувані результати навчання:**

1. Вміти розробляти комп'ютерно-інтегровані системи управління та програмно-технічні комплекси на базі мікроконтролерів і промислових логічних контролерів.
2. Розуміти суть процесів, що відбуваються в об'єктах автоматизації та вміти проводити їх аналіз і обґрунтовувати вибір структури, алгоритмів та схем керування ними на основі результатів дослідження їх властивостей.
3. Вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп'ютерні програми з використанням мов високого рівня та технологій об'єктно-орієнтованого програмування.

**Пререквізити** – «Програмування та алгоритмічні мови».

**Кореквізити** – «Проектування програмного забезпечення для спеціалізованих автоматизованих систем».

### **3. Програма навчальної дисципліни**

#### **Модуль 1.**

#### **Змістовий модуль 1. Основи синтаксису мови C++**

##### **Тема 1. Вступ до дисципліни «Програмування та алгоритмічні мови»**

Структурне програмування. Критерії якості програмного продукту. Мова програмування C++. Склад мови C++. Алфавіт C++. Поняття ідентифікатору. Ключові слова мови C++. Лексеми. Оператори. Знаки операцій.

##### **Тема 2. Типи даних мови C++**

Основні і складні типи даних. Специфікатори цілих типів даних. Типи з плаваючою крапкою. Тип void. Коментарі в C++. Структура програми в C++.

##### **Тема 3. Основні операції мови C++**

Логічний тип даних. Основні операції мови C++. Операції збільшення та зменшення на 1. Складні операції привласнення. Логічні операції. Вирази.

##### **Тема 4. Структурне програмування**

Оператори розгалуження. Умовний оператор. Тернарна операція. Оператор switch.

##### **Тема 5. Оператори циклу**

Цикл з передумовою. Цикл з постумовою. Цикл з параметром. Оператори передачі управління. Оператор break. Оператор continue. Оператор return.

##### **Тема 6. Показчики**

Визначення показчика. Показчик на функцію. Показчик на void. Способи ініціалізації показчиків. Операції з показчиками. Операція sizeof. Посилання.

##### **Тема 7. Одномірні масиви в C++**

Визначення масиву. Ініціалізація масиву. Масиви-константи. Обробка масивів.

##### **Тема 8. Розробка програм в середовищі Visual Studio**

Основні файли проекту в середовищі Visual Studio. Константи. Управляючі послідовності в мові C++. Типізація даних (особливості). Змінні. Области дії змінної.

##### **Тема 9. Багатомірні масиви в C++**

Ініціалізація багатомірних масивів. Різноманітні можливості ініціалізації багатомірних масивів. Динамічні багатомірні масиви. Обробка матриць.

#### **Модульний контроль.**

#### **Модуль 2.**

**Змістовий модуль 1. Реалізація принципів об'єктно-орієнтованого програмування в C++**

##### **Тема 10. Директиви препроцесору. Стеки**

Директива #include. Директива умовної компіляції. Директива #define. Поняття стеку.

### Тема 11. Перетворення типів. Черги. Бінарні дерева

Типи перетворення даних. Операції явного перетворення типів. Операція static\_cast. Порозрядні операції. Поняття черги. Поняття бінарного дерева.

### Тема 12. Різновиди рядків в С++

Поняття потоку. Стандартні потоки. Файлові потоки. Рядкові потоки. Ієрархія класів в С++ для підтримки потоків. Функції роботи з потоками.

### Тема 13. Робота з об'єктами класу string. С-рядки

Створення об'єкту класу string. Методи класу string. Властивості С-рядків. Функції роботи з С-рядками.

### Тема 14. Клас String^

Створення об'єкту класу String^. Методи класу String^. Розгляд прикладу роботи з об'єктами класів файлових потоків і класу String^..

### Тема 15. Структури в С++

Поняття структури в С++. Формат описання структури. Ініціалізація структури. Арифметичні операції з полями структур. Вкладені структури. Приклади роботи зі структурами.

### Тема 16. Об'єднання. Перелічування

Поняття об'єднання (union). Бітові поля. Поняття перелічування (enum). Директива using. Простір імен. Перейменування типів за допомогою typedef.

### Тема 17. Принципи об'єктно-орієнтованого програмування

Кроки підвищення абстракції коду. Поняття класу. Формат опису класу. Приховування даних, як основа об'єктно-орієнтованого програмування. Поняття конструктору. Різновиди конструкторів. Конструктор копіювання.

### Тема 18. Наслідування

Поняття інкапсуляції. Поняття поліморфізму. Поняття наслідування. Організація механізму наслідування в С++. Перевантаження операцій в С++.

### Модульний контроль.

## 4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма						заочна форма						
	усього	у тому числі					8	9	10	11	12	13	
л		п	лаб	інд	с.р.								
1	2	3	4	5	6	7	8	9	10	11	12	13	
<b>Модуль 1. Основи синтаксису мови С++</b>													
Тема 1. Вступ до дисципліни «Програмування та алгоритмічні мови»	6	1	1	1	–	3	–	–	–	–	–	–	
Тема 2. Типи даних мови С++	6	1	1	1	–	3	–	–	–	–	–	–	

Тема 3. Основні операції мови С++	6	1	1	1	-	3	-	-	-	-	-	-
Тема 4. Структурне програмування	6	1	1	1	-	3	-	-	-	-	-	-
Тема 5. Оператори циклу	6	1	1	1	-	3	-	-	-	-	-	-
Тема 6. Показчики	6	1	1	1	-	3	-	-	-	-	-	-
Тема 7. Одномірні масиви в С++	6	1	1	1	-	3	-	-	-	-	-	-
Тема 8. Розробка програм в середовищі Visual Studio	4	-	-	1	-	3	-	-	-	-	-	-
Тема 9. Багатомірні масиви в С++	6	1	1	1	-	3	-	-	-	-	-	-
Модульний контроль												
Разом за модулем 1	52	8	8	9	-	27	-	-	-	-	-	-
<b>Модуль 2. Реалізація принципів Об'єктно-орієнтованого програмування в С++</b>												
Тема 10. Директиви препроцесору. Стеки	6	1	1	1	-	3	-	-	-	-	-	-
Тема 11. Перетворення типів. Черги. Бінарні дерева	6	1	1	1	-	3	-	-	-	-	-	-
Тема 12. Різновиди рядків в С++	4	-	-	1	-	3	-	-	-	-	-	-
Тема 13. Робота з об'єктами класу string. С-рядки	6	1	1	1	-	3	-	-	-	-	-	-
Тема 14. Клас String^	6	1	1	1	-	3	-	-	-	-	-	-
Тема 15. Структури в С++	6	1	1	1	-	3	-	-	-	-	-	-
Тема 16. Об'єднання. Перелічування	8	1	1	3	-	3	-	-	-	-	-	-
Тема 17. Принципи об'єктно-орієнтованого програмування	8	1	1	3	-	3	-	-	-	-	-	-
Тема 18. Наслідування	8	1	1	3	-	3						
Модульний контроль												
Розрахункова робота	10					10						
Разом за модулем 2	68	8	8	15	-	37	-	-	-	-	-	-
1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Усього годин</b>	120	16	16	24	-	64	-	-	-	-	-	-

### 5. Темі семінарських занять

№ з/п	Назва теми	Кількість годин

## 6. Теми практичних занять

№ з/п	Назва теми	Кільк. год.
1	Вступ до дисципліни «Програмування та алгоритмічні мови»	1
2	Типи даних мови C++	1
3	Основні операції мови C++	1
4	Структурне програмування	1
5	Оператори циклу	1
6	Показчики	1
7	Одномірні масиви в C++	1
8	Розробка програм в середовищі Visual Studio	1
9	Багатомірні масиви в C++	1
10	Директиви препроцесору. Стеки	1
11	Перетворення типів. Черги. Бінарні дерева	1
12	Різновиди рядків в C++	1
13	Робота з об'єктами класу string. С-рядки	1
14	Клас String^	1
15	Структури в C++	1
16	Об'єднання. Перелічування	1
	<b>Разом</b>	<b>16</b>

## 7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Написання лінійних програм. Визначення збіжності ряду	2
2	Програма обчислення значення функції, яка обчислюється по-різному в залежності від значення аргументу	2
3	Написання програми-калькулятора	2
4	Операції з одномірними масивами	3
5	Операції з матрицями	3
6	Робота з файловими потоками і об'єктами класу String^	3
7	Класи	3
8	Наслідування	3
9	Перевантаження операцій	3
	<b>Разом</b>	<b>24</b>

## 8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Вступ до дисципліни «Програмування та алгоритмічні мови» (Тема 1)	3
2	Типи даних мови C++ (Тема 2)	3
3	Основні операції мови C++ (Тема 3)	3
4	Структурне програмування (Тема 4)	3
5	Оператори циклу (Тема 5)	3
6	Показчики (Тема 6)	3
7	Одномірні масиви в C++ (Тема 7)	3
8	Розробка програм в середовищі Visual Studio (Тема 8)	3
9	Багатомірні масиви в C++ (Тема 9)	3



10	Директиви препроцесору. Стеки (Тема 10)	3
11	Перетворення типів. Черги. Бінарні дерева (Тема 11)	3
12	Різновиди рядків в C++ (Тема 12)	3
13	Робота з об'єктами класу string. С-рядки (Тема 13)	3
14	Клас String^ (Тема 14)	3
15	Структури в C++(Тема 15)	3
16	Об'єднання. Перелічування (Тема 16)	3
17	Принципи об'єктно-орієнтованого програмування (Тема 17)	3
18	Наслідування (Тема 18)	3
	<b>Разом</b>	<b>54</b>

### 9. Індивідуальні завдання

№ з/п	Назва теми	Кільк. годин
-	Розрахункова робота. Проектування ігрових додатків	10

### 10. Методи навчання

Проведення аудиторних лекцій, практичних занять, лабораторних робіт, індивідуальні консультації (при необхідності), самостійна робота студентів.

### 11. Методи контролю

Проведення поточного контролю вивчення дисципліни, письмового модульного контролю, фінальний контроль у вигляді іспиту.

#### Розподіл балів, які отримують студенти (залік)

Поточне тестування та самостійна робота				Підсумковий тест (залік) у випадку відмови від балів поточного тестування та допуску до іспиту	
Модуль 1		Модуль 2		Сума	
T1-4	T5-9	T10-14	T15-18	100	100
25	25	25	25		

T1, ..., T18 – теми змістових модулів

### 12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
<b>Змістовний модуль 1</b>			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних	3...5	4	12...20

(практичних) робіт			
Модульний контроль	10...15	1	10...15
<b>Змістовний модуль 2</b>			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	3...5	4	12...20
Модульний контроль	10...15	1	10...15
Виконання і захист РГР (РР, РК)	16...20	1	16...20
<b>Усього за семестр</b>			<b>60...100</b>

Семестровий контроль (залік) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з 3-х питань: два теоретичних і одне практичне.

- 1 питання – 30 балів.
- 2 питання – 30 балів.
- 3 питання – 40 балів. ...

## 12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки: основи синтаксису мови C++; правила перетворення типів; як працювати об'єктами потокових класів; як працювати з об'єктами рядкових класів; як створювати власні класи.

Необхідний обсяг вмінь для одержання позитивної оцінки: використовувати механізми наслідування, інкапсуляції та поліморфізму у власних програмах; створювати власні класи та об'єкти; створювати власні програмні проекти для обробки різноманітної інформації на мові C++; стеки та організацію їх в C++; черги та організацію їх в C++; бінарні дерева та організацію їх в C++; перевантаження операцій в C++.

## 12.3 Критерії оцінювання роботи студента протягом семестру

**Задовільно (60-74).** Мати мінімум знань та умінь. Відпрацювати та захистити всі лабораторні роботи та домашнє завдання. Знати правила перетворення типів. Вміти самостійно створювати власні класи. Вміти визначати функції.

**Добре (75 - 89).** Твердо знати мінімум знань, виконати усі завдання. Показати вміння виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з обґрунтуванням рішень та заходів, які запропоновано у роботах. Вміти працювати з показниками. Вміти використовувати механізми наслідування, інкапсуляції та поліморфізму у власних програмах.

**Відмінно (90 - 100).** Повно знати основній та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Безпомилково виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з докладним обґрунтуванням рішень та

заходів, які запропоновано у роботах. Вміти працювати зі стеком та бінарними деревами. Вміти перевантажувати операції.

### Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

### 13. Методичне забезпечення

Обов'язкові складові:

- робоча програма дисципліни;
  - робочий зошит для практикуму (електронний варіант);
  - конспект лекцій, підручники (навчальні посібники), в тому числі в електронному вигляді, які за змістом повністю відповідають робочій програмі дисципліни;
  - методичні вказівки та рекомендації для виконання курсових робіт та проектів, розрахункових та розрахунково-графічних робіт, лабораторних та практичних робіт, а також рекомендації для самостійної підготовки;
  - тематики індивідуальних завдань;
  - приклади розв'язування типових задач чи виконання типових завдань;
  - питання, тести для контрольних заходів;
  - каталоги інформаційних ресурсів;
- Додаткові складові НМКД (за необхідністю):
- збірники ситуативних завдань (кейсів);
  - комп'ютерні презентації;
  - ілюстративні матеріали (плакати, таблиці тощо).

### 14. Рекомендована література

#### Базова

1. Кравець П. О. Об'єктно-орієнтоване програмування: Навчальний посібник / П.О. Кравець. — Львів: Видавництво Львівської політехніки, 2012. – 624 с.
2. C++. Основи програмування. Теорія та практика : підручник / [О.Г. Трофименко, Ю.В. Прокоп, І.Г. Швайко, Л.М. Буката та ін.] ; за ред.О.Г.Трофименко. – Одеса: Фенікс, 2010. – 544 с.

#### Допоміжна

1. C++. 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.
2. Інтернет-джерела.

### 13. Інформаційні ресурси

Сайт університету <https://www.khai.edu>

Сайт кафедри <https://k305.khai.edu>