

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Волинський національний університет
імені Лесі Українки

Факультет інформаційних технологій і математики
Кафедра комп'ютерних наук та кібербезпеки

СИЛАБУС
Вступ до програмування
підготовки бакалавра
спеціальності 125 Кібербезпека
галузі знань: 12 Інформаційні технології

Луцьк – 2020

Силабус нормативної навчальної дисципліни «Вступ до програмування» підготовки бакалавра, галузі знань 12 Інформаційні технології, спеціальності 125 Кібербезпека, освітньо-професійної програми Інформаційна безпека.

Розробник: Головін Микола Борисович, кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук та кібербезпеки

Силабус навчальної дисципліни, затверджено на засіданні кафедри

Комп'ютерних наук та кібербезпеки
протокол № 3 від 05.10.2020р.

Завідувач кафедри:



Сачук Ю.В.

1. Опис навчальної дисципліни

Характеристика навчальної дисципліни подається згідно з навчальним планом і представляється у вигляді таблиці 1.

Таблиця 1

Найменування показників	Галузь знань, спеціальність, освітня програма, освітній ступінь	Характеристика навчальної дисципліни
		денна форма навчання
Денна форма навчання	12 - Інформаційні технології 125 - Кібербезпека Бакалавр	нормативна
Кількість годин/кредитів 150/5		Рік підготовки другий
ІНДЗ: є		Семестр 3
		Лекції 30 год.
		Практичні 0 год. Лабораторні 38 год.
		Самостійна робота 74 год. Консультації 8 год.
		Форма контролю: екзамен

2. Анотація курсу

Силабус навчальної дисципліни «Вступ до програмування» складена відповідно до освітньо-професійної програми підготовки ОКР «Бакалавр», підготовки бакалавра галузі знань 12 - Інформаційні технології, спеціальність 125 - Кібербезпека, освітньо-професійна програма - Інформаційна безпека.

Предметом вивчення навчальної дисципліни «Вступ до програмування» є:

- основи мови C++ та стандартні алгоритмічні конструкції, що можуть бути реалізовані цією мовою;
- технології неспітної деталізації алгоритмів та модульного, по-блокового програмування на прикладах простих консольних та Windows додатків написаних мовою C++;
- процедурне, об'єктно-орієнтоване, візуальне програмування, як ілюстрація сучасних технологічних парадигм в створенні сучасного програмного забезпечення.

Місце навчальної дисципліни в системі професійної підготовки фахівця: дисципліна «Вступ до програмування» належить до нормативних фундаментальних дисциплін і є базовою для вивчення спеціальних курсів студентами спеціальності 125 – Кібербезпека. Це одна із *основних складових* циклу *професійної* підготовки. Передують вивченню курси методи обробки даних та вступ до фаху.

Програма навчальної дисципліни складається з наступних **змістових модулів**.

1. Локальна модульність стандартних алгоритмічних конструкцій, що можуть бути реалізовані мовою C++. Основи мовних конструкцій, структури даних, вводу/виводу.
2. Модульність програмного коду, сучасні технологічні особливості мовних засобів, що спрощують процес написання програм.

Метою викладання навчальної дисципліни «Вступ до програмування» є вивчення основних, найбільш загальних сучасних модульних підходів створення коду на прикладах програм написаних мовою C++.

Основними завданнями вивчення дисципліни «Вступ до програмування» є:

- Сформувати у студентів причинно-наслідкове, абстрактно-логічне мислення, яке є основою наукового, матеріалістичного менталітету, спираючись, з одного боку, на алгоритмічну складову завдань модульного програмування, а з іншого боку, на відлагодження програмного коду в ході виконання лабораторних робіт.
- Вивчити основи мови C++ та стандартні алгоритмічні конструкції, що можуть бути реалізовані цією мовою на прикладах простих консольних та Windows додатків написаних мовою C++;
- Засвоїти фундаментальні технологіями створення програмних продуктів, які завжди базуються на двох ментальних шляхах: несхідній деталізації алгоритмів та на модульному, блоковому створенні програмних конструкцій.
- Ознайомити студентів з прийомами процедурного, об'єктно-орієнтованого та візуального програмування, як ілюстрація сучасних технологічних парадигм в створенні програмного забезпечення.

Згідно з вимогами освітньо-професійної програми студенти повинні:

знати

- технології створення та відлагодження програмних конструкцій;
- структури даних, що характерні для мови C++;
- основні конструкції мови програмування;
- алгоритмічні конструкції керування ходом виконання програм;
- основи процедурного, об'єктно-орієнтованого та візуального програмування

вміти :

- мислити причинно-наслідково, абстрактно-логічно застосовуючи методики несхідної деталізації алгоритму і модульного програмування;
- застосовувати в ході написання програмного коду алгоритмічні конструкції керування ходом виконання програм;
- використовувати сучасні мовні конструкції процедурного, об'єктно-орієнтованого та візуального програмування;
- відлагоджувати програмні конструкції та доводити програму до робочого стану. На вивчення навчальної дисципліни відводиться 150 годин / 5 кредитів ECTS.

Форма контролю – екзамен.

3. Компетенції

Інтегральна компетентність (ІНТ)

- Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі забезпечення інформаційної безпеки і/або кібербезпеки, що характеризується комплексністю та неповною визначеністю умов.

Загальні компетентності (ЗК)

ЗК5. Здатність до пошуку, оброблення та аналізу інформації.

Фахові компетентності (ФК)

ФК 2 Здатність до використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки.

Програмні результати навчання

1. Вирішувати задачі аналізу програмного коду на наявність можливих загроз. (ПРН 53)
2. Забезпечувати процеси захисту та функціонування інформаційно- телекомунікаційних (автоматизованих) систем на основі практик, навичок та знань, щодо структурних (структурно-логічних) схем, топології мережі, сучасних архітектур та моделей захисту електронних інформаційних ресурсів з відображенням взаємозв'язків та інформаційних потоків, процесів для внутрішніх і віддалених компонент. (ПРН 17)

4. Інформаційний обсяг навчальної дисципліни Структура навчальної дисципліни

Структура навчальної дисципліни представляється у вигляді таблиці 2.

Таблиця 2.

Назви змістових модулів і тем	Кількість годин					
	Усього	у тому числі				
		Лек.	Практ. (Семін.)	Лаб.	Самос. роб.	Конс.
1	2	3	4	5	6	7
Модуль 1. Локальна модульність стандартних алгоритмічних конструкцій, що можуть бути реалізовані мовою C++. Основи мовних конструкцій, структури даних, вводу/виводу						
Тема 1. Алгоритм. Програма. Етапи розв'язування прикладної задачі з використанням ЕОМ. Низхідний спосіб деталізації проблеми. Структурний модульний спосіб її вирішення.	5	1			4	
Тема 2. Класифікація мов програмування. Транслятор. Інтерпретація та компіляція. Інтегроване середовище програмування.	5	1			4	
Тема 3. Особливості та призначення мови C++. Базові поняття. Основні типи даних (вбудовані типи). Модульна структура програми на мові програмування C++.	7	1		2	4	
Тема 4. Ввод з консолі та вивід. Присвоєння. Пріоритет операцій. Арифметичні вирази. Стандартні функції.	7	1		2	4	
Тема 5. Розгалуження та цикли, як локальні модульні компоненти структури програми. Керування ходом виконання програми.	10	2		2	6	
Тема 6. Укрупнені локальні модульні програмні конструкції на мові C++: розгалуження в розгалуженні та розгалуження в циклі	10	2		2	5	1
Тема 7. Робота з файлами. Стандартні функції роботи з файлами. Приклади програм: створення, читання, дописування, виправлення файлу.	10	2		2	5	1
Тема 8. String - шаблонний строковий клас. Огляд стандартних функцій. Приклади програм, що реалізують: пошук, заміну в тексті; розрізання речення на слова; шифрування перестановкою та зміщенням коду знаків.	12	2		4	5	1
Тема 9. Одномірні і багатовимірні масиви. Динамічна пам'ять. Програми пошуку суми, середнього та максимального. Сортування. Шифрування перестановкою в масиві	12	2		4	5	1
Усього годин за перший модуль	78	14		18	42	4
Модуль 2. Модульність програмного коду, сучасні технологічні особливості мовних засобів, що спрощують процес написання програм.						
Тема 10. Функції користувача. Модульні конструкції побудовані на основі власних функцій і бібліотек.	8	2		2	4	
Тема 11. Побудова графіків 2d, 3d математичних функцій. Елементи комп'ютерної анімації. Моделювання фізичних процесів.	8	2		2	4	

Тема 12. Нова сучасна якість модульного підходу до програмування. Об'єктно – орієнтоване програмування. Модулі, класи, об'єкти. Приклади використання в програмах.	8	2		2	4	
Тема 13. Модульність в реалізації візуального програмного інтерфейсу. Вікно форми. Вікно інспектору об'єктів. Вікно коду програми. Палітра компонентів.	8	2		2	4	
Тема 14. Інтерфейс Windows на прикладі програми "Калькулятор". Стандартні візуальні компоненти інтерфейсу: форма, панель, кнопки, однорядкові редактори.	9	2		2	4	1
Тема 15. Інтерфейс Windows на прикладі програми "Блокнот". Робота з файлами. Компоненти: багаторядковий редактор, основне меню.	9	2		2	4	1
Тема 16. Ускладнення та трансформація програми "Блокнот" в оболонку для шифрування. Компоненти: список вибору, закладки, однопозиційні та багатопозиційні перемикачі.	10	2		4	3	1
Тема 17. Модульна формалізація простих алгоритмів шифрування в вигляді функцій користувача. Впровадження цих функцій в програми кодування та шифрування в сучасному інтерфейсі Windows.	12	2		4	5	1
Усього годин за другий модуль	72	16		20	32	4
Усього годин	150	30		38	74	8

Теми лабораторних занять

№ з/п	Тема	Кількість годин
1	Особливості та призначення мови C++. Базові поняття. Основні типи даних (вбудовані типи). Модульна структура програми на мові програмування C++.	2 год
2	Ввод з консолі та вивід. Присвоєння. Пріоритет операцій. Арифметичні вирази. Стандартні функції.	2 год
3	Розгалуження та цикли, як локальні модульні компоненти структури програми. Керування ходом виконання програми.	2 год
4	Укрупнені локальні модульні програмні конструкції на мові C++: розгалуження в розгалуженні та розгалуження в циклі	2 год
5	Робота з файлами. Стандартні функції роботи з файлами. Приклади програм: створення, читання, дописування, виправлення файлу.	2 год
6	String - шаблонний строковий клас. Огляд стандартних функцій.. Приклади програм, що реалізують: пошук, заміну в тексті; розрізання речення на слова; шифрування перестановкою та зміщенням коду знаків.	4 год
7	Одномірні і багатовимірні масиви. Динамічна пам'ять. Програми пошуку суми, середнього та максимального. Сортування. Шифрування перестановкою в масиві	4 год

8	Функції користувача. Модульні конструкції побудовані на основі власних функцій і бібліотек.	2 год
9	Побудова графіків 2d, 3d математичних функцій. Елементи комп'ютерної анімації. Моделювання фізичних процесів.	2 год
10	Нова сучасна якість модульного підходу до програмування. Об'єктно – орієнтоване програмування. Модулі, класи, об'єкти. Прикладивикористання в програмах.	2год
11	Модульність в реалізації візуального програмного інтерфейсу. Вікно форми. Вікно інспектору об'єктів. Вікно коду програми. Палітра компонентів.	2 год
12	Інтерфейс Windows на прикладі програми "Калькулятор". Стандартні візуальні компоненти інтерфейсу: форма, панель, кнопки,однорядкові редактори.	2 год
13	Інтерфейс Windows на прикладі програми "Блокнот". Робота з файлами. Компоненти: багаторядковий редактор, основне меню.	2 год
14	Ускладнення та трансформація програми "Блокнот" в оболонку для шифрування. Компоненти: список вибору, закладки, однопозиційніта багатопозиційні перемикачі.	4 год
15	Модульна формалізація простих алгоритмів шифрування в вигляді функцій користувача. Впровадження цих функцій в програмикодування та шифрування в сучасному інтерфейсі Windows.	4 год
	Разом	38 год

5. Завдання для самостійного опрацювання

Самостійна робота студента складається з кількох частин.

1. Опрацювання лекційного матеріалу, що дублюється і розширюється матеріалами інформаційної частини відповідного дистанційного курсу на платформі MOODLE.

2. Підготовка до лабораторних занять, яка, зокрема, полягає в виконанні тестових завдань, що викладені в достатній кількості на дистанційній платформі MOODLE і підтримують кожен тему курсу. Ці завдання орієнтовані на дві модальності засвоєння матеріалу: а). службові слова мови програмування, їх формат, параметри та функціональні особливості застосування; б). характерні для поточної теми стандартні алгоритмічні конструкції, їх програмні реалізації.

3. Виконання індивідуальних програмних проектів для студентів, які претендують на високі оцінки. Пошук та засвоєння додаткових матеріалів необхідних для реалізації проекту. Маються на увазі матеріали, які дотичні до основного курсу, однак не розглядаються в курсі лекцій.

4. Систематизація вивченого матеріалу перед іспитом.

6. Види індивідуальних науково-дослідних завдань

Передбачається виконання індивідуальних програмних проектів. Тематика проектів безпосередньо пов'язана з курсом і стосується роботи з файлами, захисту інформації методом її шифрування та приховування. Також можна вибрати тематику проекту пов'язану з комп'ютерною фізикою, зокрема, з моделюванням процесів і явищ фізики.

7. Розподіл балів та критерії оцінювання

Поточний контроль (макс = 40 балів)																	Модульний контроль		Загальна кількість балів		
Модуль 1										Модуль 2			Модуль 3								
Змістовий модуль 1					Змістовий модуль 2							ІНДЗ	МКР1	КР2							
Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	10	30	30	100	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17					
2	2	3	3	3	2	2	2	2	2	2	3	2	2	2	3	3	2				

Підсумковий контроль успішності навчання формується двома складовими: **поточним та модульним контролем**. Оцінювання знань здійснюється із використанням **100** бальної шкали.

Поточний контроль. Виконання пакетів завдань в кожній з **15** тем (починаючи з третьої) по першому і другому змістовному модулю дає по **2** бали. Всього на тестах в поточному контролі можна отримати максимально **30** балів. Інформація стосовно виконань завдань поточного контролю знаходиться в журналі відповідного MOODLE курсу. Оцінка за виконаний тест лабораторного заняття вказує на ступінь **ознайомлення** студента з новим матеріалом відповідної теми. Використання інформаційної складової курсу під час виконання тестів поточного контролю допускається. Обмеження по часу виконання пакету завдань відсутнє. Можливості стосовно **застосування отриманих** знань на практиці оцінюються проектом, який виконує студент. Проект вважається індивідуальним завданням (**ІНДЗ**). За проект в поточному оцінюванні студент може отримати максимум **10** балів.

Модульний контроль складається з двох модульних контрольних робіт (**МКР**), що відповідні першому і другому змістовному модулю. Пакет завдань МКР містить завдання з відповідних тем змістовного модуля. Завдання з пакету вибираються випадковим чином. Кожна з модульних контрольних робіт може бути оцінена максимум на **30** балів. Всього за дві МКР можна набрати максимум **60** балів. Використання інформаційної складової курсу під час виконання тестів модульного контролю не допускається. Існує обмеження по часу виконання пакету завдань. Дається одна спроба на виконання кожної МКР. Інформація стосовно виконань завдань модульного контролю знаходиться в журналі відповідного MOODLE курсу. Вважається, що студенти готуючись МКР **закріплюють** новий матеріал. Після МКР студенти мають вільно володіти матеріалом і застосовувати його на практиці з мінімальним використанням довідкової інформації.

Екзамен проводиться в тестовій формі. На екзамені максимально можна набрати **60** балів. Екзамен здають студенти незадоволені своєю оцінкою за курс, а також ті, що набрали менше 75 балів. Пакет екзаменаційних завдань формується з всієї сукупності завдань курсу. Завдання з пакету вибираються випадковим чином. Час проведення екзамену обмежений. Дається одна спроба на виконання пакету екзаменаційних завдань. Використання інформаційної складової курсу на екзамені забороняється. Студент, що йде на екзамен обнуляє результати МКР.

Шкала оцінювання (національна та ECTS)

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену,	для заліку
90 – 100	A	Відмінно	Зараховано
82 – 89	B	Добре	
75 - 81	C		
67 -74	D		
60 - 66	E	Задовільно	Незараховано (з можливістю повторного складання)
1 – 59	F x	Незадовільно	

8. Методичне забезпечення та рекомендована література

Перелік методичних видань для забезпечення вивчення навчальної дисципліни:

Дисципліна має підтримку дистанційного курсу «Вступ до програмування» на платформі MOODLE <http://194.44.187.60/moodle/course/view.php?id=1306> . В цьому курсі крім інформаційної частини є потужна інтерактивна складова в вигляді 20 пакетів тестових завдань, що сприяють систематизації, усвідомленню та закріпленню нового матеріалу по кожній темі. Також безпосередньо в курсі можна запускати та відлагоджувати програми online.

Література

1. Бьярне Страуструп Программирование: принципы и практика с использованием C++, 2-е изд. : Пер. с англ. - М. : ООО "И.Д. Вильямс", 2016. - 1328 с.
2. Шлее М. Qt 5.10. Профессиональное программирование на C++. — СПб.: БХВ-Петербург, 2018. — 1072 с.
3. Скотт Мейерс Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14. : Пер. с англ. - М. : ООО "И.Д. Вильямс", 2016. - 304 с.
4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб.: Питер, 2015. — 368 с.
5. Галовиц Я. C++17 STL Стандартная библиотека шаблонов. — СПб.: Питер, 2018. — 432 с.
6. Сиддхартха Рао Освой самостоятельно C++ по одному часу в день, 8-е изд.: Пер. с англ. — СПб.: ООО "Альфа-книга", 2017. — 752 с
7. Джордж Хайнеман, Гэри Поллис, Стэнли Селков Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО "Альфа-книга", 2017. — 432 с.
8. Стивен Прата Язык программирования C Лекции и упражнения. 6-е изд. : Пер. с англ. ООО "И.Д. Вильямс", 2015. — 928 с.
9. Никехин А.А. Основы C++ для моделирования и расчетов. Учебное пособие. СПб.: НИУ ИТМО, 2014. – 106 с.
10. Шилдт Герберт C++ для начинающих. Серия «Шаг за шагом»/ Шилдт Г.; пер. с англ. -М.: ЭКОМ Паблишерз, 2013. - 640 с.
11. Стенли Б.Липпман, Жози Лажойе, Барбара Э. Му Язык программирования C++. Базовый курс, 5-е изд. : Пер. с англ. М.: ООО "И.Д. Вильямс", 2014. — 1120 с.
12. Энтони Уильямс Параллельное программирование на C++ в действии. Практика разработки многопоточных программ. Пер. с англ. Слинкин А.А. – М.: ДМК Пресс, 2012. – 672с.:
13. Трофименко О.Г. C++. Основы 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.
14. Динман М. И. C++. Освой на примерах. — СПб.: БХВ-Петербург, 2006. - 384 с.
15. Пахомов Б. И. C/C++ и MS Visual C++ 2008 для начинающих. -2009. — 624 с.
16. Доїбуш, Г. Ф.
17. Visual C++ на примерах / Г. Ф. Довбуш, А. Д. Хомоненко / Под ред. проф. А. Д.Хомоненко. — СПб.: БХВ- Петербург, 2007. - 528 с.
18. Том Арчер, Эндрю Уайтчепел Visual C++ .NET. Библия пользователя.: Пер. с англ. —М.: Издательский дом "Вильямс", 2005. — 1216 с.
19. Дал У., Дейкстра Э., Хоар К. Структурное программирование. – М.: Мир, 1975. – 246 с.
20. Хьюз Дж., Митчом Дж. Структурный подход к программированию. – М.: Мир, 1980. –276 с.
21. Культин Н.Б. Основы программирования в Microsoft® Visual C++ 2010. — СПб.: БХВ-Петербург, 2010. — 384 с.: ил.

22. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++: Учебный курс, Харьков: Фолио; М: ООО «Издательство АСТ», 2001. — 500 с.

23. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.

Політика курсу

Політика цього курсу визначається місцем його в системі підготовки студентів. Це нормативний курс з циклу професійної підготовки. Як викладач, для забезпечення знань, умінь та навичок студентів та здобуття потрібних компетенцій, я підготував та ввів в дію дистанційний курсу «Вступ до програмування» на платформі MOODLE <http://194.44.187.60/moodle/course/view.php?id=1306>. Інформаційна частина цього курсу повністю охоплює матеріал курсу. В цьому курсі є також потужна інтерактивна складова в вигляді 20 пакетів тестових завдань. Кожен пакет містить кілька десятків завдань. Виконання завдань по кожній темі сприяє систематизації, усвідомленню та закріпленню нового матеріалу. Безпосередньо в курсі можна відлагоджувати, модифікувати та запускати програми аплікації online. Це зручно, адже дозволяє освоювати курс з будь-якого робочого місця підключеного до інтернету.

Робота зі студентами передбачена, як в традиційному режимі, коли студенти присутні на заняттях очно і працюють на лекціях, практичних, лабораторних та консультаціях, так і в повністю дистанційному режимі. Зрозуміло, що можливі і проміжні протоколи роботи з студентами, коли частина занять проводиться дистанційно, а частина очно.

Оцінювання. Протягом семестру для перевірки знань студентів та контролю за самостійною роботою студента застосовують лабораторні роботи. Вважається, що процес освоєння матеріалу має кілька градацій.

На першому етапі, назвемо його **ознайомлювально орієнтаційним**, студенту необхідно прослухати лекцію або прочитати відповідну інформаційну частину в moodle курсі і виконати тестові завдання теми. Це невелика порція матеріалу і контроль за часом виконання завдань і використанням інформаційної частини курсу не проводиться. Оцінюється ця частина роботи в 2 бали в поточному контролі. На протязі курсу в 15 темах можна набрати **30 балів**.

На другому, етапі засвоєння матеріалу студент має практично застосувати отримані знання. Назвемо його **структурно систематизаційним**. Тобто відбувається систематизація та структурування знань в контексті практичного питання «для чого цей програмний засіб?». Цей етап триває на протязі всього курсу в процесі реалізації власного проекту. На цьому етапі можна набрати **10 балів**. Передбачається представлення повного тексту програми, та захист проекту. В процесі захисту студент повинен сказати коротко, по кілька речень по кожному наступному пункту, тема роботи, огляд подібних робіт, актуальність напрямку роботи, мета роботи, завдання роботи, математична модель закладена в програму, алгоритм реалізації задуму, вибір інструменту реалізації задуму (вибір мови програмування), реалізація програми, обговорення ключових цікавих моментів реалізації програми, апробація програми та обговорення результатів, висновки.

Останній третій етап, назвемо його **закріплювально завершальним**, пов'язаний з остаточним закріпленням матеріалу курсу в процесі підготовки до модульних контрольних робіт або екзамену. Виконання двох контрольних модульних робіт оцінюється по **30 балів**, екзаменаційної роботи **60 балів**. Тут перевіряються великі порції матеріалу проводиться контроль за часом виконання завдань. Використанням інформаційної частини курсу заборонено. Оцінюється ця частина роботи в модульному або екзаменаційному контролі.

Відвідування. Зазначалось, що курс може бути проведений, як повністю очний, так і повністю дистанційний. Зрозуміло, що можуть бути реалізовані будь-які комбіновані варіанти. Однак в повністю дистанційному курсі виникає проблема доброчесності. Ніколи неможна бути до кінця впевненим, що студент сам виконує завдання і взагалі є причетним до будь-яких результатів зворотного зв'язку курсу. Тому в політиці курсу є вимога очної присутності на модульних контрольних заходах. Та вимога очного захисту власного проекту.

Щодо академічної доброчесності. Під час виконання модульної, підсумкової контрольної роботи вважається шахрайством заглядання в будь-які інформаційні ресурси розташовані в додаткових вікнах, закладках екрану, на мобільному телефоні, шпаргалках, конспекті, підручнику. Заборонено також підказувати та слухати підказки. Проект студента повинен перевірятись на плагіат і має бути захищений персонально.

Програмування є галуззю знань, що постійно трансформується та модифікується. Курс «Вступ до програмування» теж змінюється. Покращується інформаційна частина курсу, додаються нові завдання. Тому в процесі поточного розгляду матеріалів приймаються всілякі побажання до інформаційної частини курсу, в процесі поточного контролю передбачає обговорення тестових завдань та їх корекція. На етапі модульного контролю будь-які претензії студентів, як до інформаційної частини курсу, так і до завдань не приймаються.

Академічна доброчесність базується на згоді усіх учасників процесу навчання: викладач та студенти, дотримуватися правил та виконувати узяті зобов'язання.

9. Перелік питань до екзамену

Екзамен проводиться автоматизовано через тестову частину курсу. Пакет екзаменаційних завдань і модульних контрольних робіт формується з числа завдань, що розглядалися в процесі поточного контролю. Всі ці завдання виконувались студентами. Стосовно всіх завдань приймалися зауваження і побажання. Ускладнення ситуації екзамену або модульної контрольної роботи полягає в забороні користування інформаційними ресурсами, обмежені часу виконання завдань, в великій кількості цих завдань, широкому охоплені матеріалу, що контролюється та в випадковому виборі конкретних завдань.