

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«ПРИДНІПРОВСЬКА ДЕРЖАВНА АКАДЕМІЯ БУДІВНИЦТВА ТА АРХІТЕКТУРИ»**

КАФЕДРА прикладної математики та інформаційних технологій  
(повна назва кафедри)

**«ЗАТВЕРДЖУЮ»**  
Проректор з науково-педагогічної та навчальної роботи  
Р. Б. Папірник

« 15 » \_\_\_\_\_ 20 19 року

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Об'єктно-орієнтоване програмування**

(назва навчальної дисципліни)

спеціальність \_\_\_\_\_ 122 «Комп'ютерні науки» \_\_\_\_\_

(шифр і назва спеціальності)

освітньо-професійна програма \_\_\_\_\_ «Комп'ютерні науки» \_\_\_\_\_

(назва освітньої програми)

освітній ступінь \_\_\_\_\_ бакалавр \_\_\_\_\_

(назва освітнього ступеня)

форма навчання \_\_\_\_\_ денна \_\_\_\_\_

(денна, заочна, вечірня)

розробник \_\_\_\_\_ Власенко Юрій Євгенович \_\_\_\_\_

(прізвище, ім'я, по батькові)

**1. АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Потреба в написанні програм з'явилась разом з появою комп'ютерної техніки. Поява мов програмування дало можливість писати програми в декілька тисяч строк. Однак мова програмування, легко зрозуміла в коротких програмах, коли справа стосувалася великих програм, ставала некерованою. Тоді і з'явилося структурне програмування, суттю якого є можливість розбиття програми на складові її елементи. Хоча структурне програмування принесло видатні результати, навіть воно виявлялося неспроможним тоді, коли програма досягала певної довжини. Щоб написати більш складну програму, необхідний був новий підхід до програмування.

В результаті були розроблені принципи об'єктно-орієнтованого програмування. ООП акумулює кращі ідеї, втілені в структурному програмуванні, і поєднує їх з новими концепціями, які дозволяють оптимально організувати програми. Об'єктно-орієнтоване програмування дозволяє розкласти проблему на складові частини. Кожна складова стає самостійним об'єктом, який містить свої власні коди і дані, які відносяться до цього об'єкту. У цьому випадку вся процедура в цілому спрощується, і програміст отримує можливість оперувати з набагато більшими за обсягом програмами. На даний час саме ця парадигма використовується в переважній більшості промислових проектів.

Курс ООП дозволить студенту розібратись і освоїти основні складові об'єктно-орієнтованого програмування – інкапсуляцію, поліморфізм та спадкування і використовувати їх при написанні своїх програм.

## 2. ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

	Години	Кредити	Семестр	
			II	III
Всього годин за навчальним планом, з них:	270	9	120	150
<b>Аудиторні заняття, у т.ч:</b>	96		36	60
лекції	52		22	30
лабораторні роботи	30		14	16
практичні заняття	14			14
<b>Самостійна робота, у т.ч:</b>	174		84	90
підготовка до аудиторних занять	36		16	20
підготовка до контрольних заходів	24		14	10
опрацювання розділів програми, які не викладаються на лекціях	69		54	15
виконання курсової роботи	15			15
підготовка до екзамену	30	1		30
<b>Форма підсумкового контролю</b>			залік	екзамен

## 3. СТИСЛИЙ ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**Мета дисципліни** знайомство студентів з базовими концепціями ООП (інкапсуляція, поліморфізм, спадкування), набуття навиків використання мови програмування, що підтримує методологію ООП (C++), знайомство зі стандартними алгоритмами та шаблонами бібліотеки STL.

Курс «Об'єктно-орієнтоване програмування» дає основу для подальшого вивчення усього циклу комп'ютерних дисциплін та самостійного оволодіння будь-якими мовами програмування, що дозволяють застосовувати методологію ООП. Цей курс відкриває перед студентом двері у світ сучасних інформаційних технологій, як для професійного розробника.

**Завдання дисципліни** освоєння об'єктно орієнтованого підходу до програмування

**Пререквізити дисципліни** «Алгоритмізація та програмування».

**Постреквізити дисципліни** «Організація баз даних і знань», «Програмування на мові VBA», «Технологія створення програмних продуктів».

**Компетентності ІК.** Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у сфері комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

**ЗК-1.** Здатність до абстрактного мислення, аналізу та синтезу, **ЗК-8.** Здатність генерувати нові ідеї (креативність).

**СК-3.** Здатність до побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення та 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, **СК-8.** Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: структурного, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.

**РН-1.** Здобувати систематичні знання в галузі комп'ютерних наук, аналізувати проблеми з точки зору сучасних наукових парадигм, осмислювати і робити обґрунтовані висновки з наукової і навчальної літератури та результатів експериментів, **РН-2.** Реалізовувати засвоєні поняття, концепції, теорії та методи в інтелектуальній і практичній діяльності в галузі комп'ютерних наук, осмислювати зміст і послідовність застосування

способів виконання дій, узагальнювати і систематизувати результати робіт, **РН-3**. Професійно спілкуватись державною та іноземними мовами, розробляти державною та іноземними мовами документацію на системи, продукти і сервіси інформаційних технологій, читати, розуміти та застосовувати технічну документацію українською та іноземними мовами в професійній діяльності, **РН-10**. Аналізувати проблемні ситуації, ставити собі певні цілі щодо розв'язання професійних задач і свідомо добиватися їх реалізації, вибирати шлях для майбутніх дій, визначати засоби, потрібні для досягнення мети, приймати рішення, **РН-14**. Використовувати формальні моделі алгоритмів та обчислюваних функцій, встановлювати розв'язність, часткову розв'язність та нерозв'язність алгоритмічних проблем, проектувати, розробляти та аналізувати алгоритми, оцінювання їх ефективності та складності.

**Заплановані результати навчання** У результаті вивчення навчальної дисципліни студент повинен

**знати:**

- особливості об'єктно-орієнтованої мови програмування (C++);
- основні концепції ООП (інкапсуляція, поліморфізм, спадкування);
- можливості стандартної бібліотеки шаблонів (STL);
- основні (та найчастіше вживані) шаблони проектування.

**вміти:**

- описувати класи, їх атрибути і методи;
- розуміти призначення та використовувати конструктори, деструктори, переважані функції та оператори;
- описувати і використовувати статичні атрибути та методи класів;
- описувати та використовувати дружні функції класів;
- використовувати при розробці класів інкапсуляцію, поліморфізм та спадкування (у т.ч. множинне);
- застосовувати при створенні програм деякі стандартні шаблони STL (вектор, стек, алгоритми);
- застосовувати при розробці програм шаблони проектування.

**Методи навчання** : практичний, наочний, словесний, робота з книгою.

**Форми навчання** : фронтальні, групові, аудиторні, позааудиторні.

#### 4. СТРУКТУРА (ТЕМАТИЧНИЙ ПЛАН) ДИСЦИПЛІНИ

Назва змістових модулів і тем	Кількість годин, у тому числі				
	усього	л	п	лаб.	с.р
<b>II семестр</b>					
<b>Змістовий модуль 1. Інкапсуляція</b>					
<b>Комп'ютер, який ми програмуємо.</b> Робота з пам'яттю комп'ютера. Біти та байти пам'яті як віддзеркалення цифр двійкових чисел. Числове кодування текстових символів. Адреса змінної і вказівник. Побітові операції з цілими двійковими числами.	<b>31</b>	<b>6</b>		<b>4</b>	<b>21</b>
<b>Функції користувача. Використання функцій та масивів в C++.</b> Функції як модульний принцип побудови програм. Inline функції.	<b>29</b>	<b>4</b>		<b>4</b>	<b>21</b>

Функції із змінною кількістю параметрів. Перевантаження функцій. Масиви структур та масиви рядків. Функції в C++: передача функції, як параметра іншої функції за допомогою вказівника. Особливості роботи з масивами, як параметрами функцій.					
<b>Вказівники та посилання.</b> Вказівники та посилання. Оператори (*) та (&). Вказівники та масиви. Вказівники та посилання на функцію. Символьна інформація та рядки. Масиви структур та масиви рядків.	29	6		2	21
<b>Типи даних, що визначаються користувачем.</b> Бітові поля. Об'єднання. Динамічні структури даних.	31	6		4	21
<b>Разом за змістовим модулем 1</b>	<b>120</b>	<b>22</b>		<b>14</b>	<b>84</b>
<b>III семестр</b>					
<b>Змістовий модуль 2. Поліморфізм та спадкування</b>					
<b>Класи та абстрагування даних.</b> Принципи об'єктної орієнтації. Визначення та реалізація структур та класів. Управління доступом до функцій-елементів та даних-елементів класу. Використання конструкторів та деструкторів. Відокремлення інтерфейсу від реалізації класу.	23	6	4	4	9
<b>Класи: композиція та динамічне управління об'єктами.</b> Константні об'єкти та функції-елементи. Дружні функції та дружні класи. Динамічне розподілення пам'яті за допомогою операторів new і delete. Абстракція даних та приховання інформації.	21	4	4	4	9
<b>Спадкування. Віртуальні функції, поліморфізм.</b> Базові та похідні класи. Множинне спадкування. Віртуальні функції. Абстрактні та конкретні класи. Поліморфізм. Віртуальні деструктори.	21	6	2	4	9
<b>Шаблони, родові функції та класи, обробка виключних ситуацій.</b> Поняття базового набору операцій. Типи даних, з якими оперує родова функція та клас. Використання ключового слова template. Що таке обробка виключних ситуацій. Використання ключових слів try, catch та throw при обробці виключних ситуацій.	21	8	2	2	9
<b>Бібліотека стандартних шаблонів. Класи контейнери. Вектори, списки, алгоритми.</b> Знайомство з бібліотекою. Що таке контейнери, ітератори, аллокатори, предикати. Приклади використання векторів, списків, алгоритмів.	19	6	2	2	9
<b>Разом за змістовим модулем 2</b>	<b>105</b>	<b>30</b>	<b>14</b>	<b>16</b>	<b>45</b>
<b>Змістовий модуль 3. Курсова робота «Побудова класу Rectangle»</b>					
<b>Виконання курсової роботи</b>					
Аналіз поставленої задачі, вибір об'єктів для рішення.	3				3

Складовий состав об'єктів, назва, методи, що вони повинні виконувати, данні члени.	3				3
Програмні алгоритми методів та їх відладка.	3				3
Програмний алгоритм всієї роботи.	3				3
Оформлення звіту та його захист.	3				3
<b>Разом за змістовим модулем 3</b>	<b>15</b>				<b>15</b>
<b>Підготовка до екзамену</b>	<b>30</b>				<b>30</b>
<b>Усього годин</b>	<b>270</b>	<b>52</b>	<b>14</b>	<b>30</b>	<b>174</b>

### 5. ЛЕКЦІЙНИЙ КУРС

№ зан.	Тема занять	Кількість годин
1-3	Комп'ютер, який ми програмуємо.	6
4,5	Функції користувача. Використання функцій та масивів в C++.	4
6-8	Вказівники та посилання	6
9-10	Типи даних, що визначаються користувачем.	6
11-13	Класи та абстрагування даних.	6
14,15	Класи: композиція та динамічне управління об'єктами.	4
16-19	Спадкування. Віртуальні функції. поліморфізм.	6
20-23	Шаблони, родові функції та класи, обробка виключних ситуацій.	8
24-26	Бібліотека стандартних шаблонів. Класи контейнери. Вектори, списки, алгоритми.	6
Усього годин		52

### 6. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

№ зан.	Тема занять	Кількість годин
1-3	Робота з позиційними системами числення.	6
4-6	Робота з багатовимірними масивами.	6
7	Робота з простором імен. Визначення простору імен. Оператор дозволу області видимості. Вкладені простори імен.	2
Усього годин		14

### 7. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ зан.	Тема занять	Кількість годин
1-2	Виконання програм простої структури. Робота з введенням та виведенням даних	4
3-5	Операції та вирази. Операції з цілими числами.	6
6-8	Функції. Передача масивів в функцію та із функції. Робота з динамічною пам'яттю.	6
9-15	Робота з двовимірним масивом як з об'єктом. Реалізація суми та різниці об'єктів масивів.	14
Усього годин		30

### 8. САМОСТІЙНА РОБОТА

№ п/п	Вид роботи / Назва теми	Кількість годин
1	підготовка до аудиторних занять	36
2	підготовка до контрольних заходів	24
3	опрацювання розділів програми, які не викладаються на лекціях	69
	– електронна пам'ять - «технологічний простір» для обробки представлених в числовій формі даних	3
	– біти пам'яті як відображення цифр двійкових чисел	1
	– шістнадцяткове уявлення двійкових чисел	2
	– числове кодування текстових символів	2
	– кодування зображень текстових символів	3
	– адреса числа в пам'яті - це теж число	3
	– адреса і вказівник	3
	– текстові рядки як послідовність двійкових чисел і деякі операції з ними	4
	– команди обробки чисел - це теж числа	3
	– пам'ять з довільним доступом і пам'ять тільки для читання	3
	– реєстри і порти пристроїв, що входять в склад комп'ютера	4
	– побітові операції з цілими двійковими числами	3
	– короткий огляд деяких бібліотек. Загальні зауваження	3
	– стандартні математичні функції	5
	– функції класифікації і перетворення символів	7
	– функції для роботи з блоками пам'яті	7
	– функції обробки текстових рядків	5
	– визначення простору імен	3
	– оператор дозволу області видимості	2
	– вкладені простори імен	3
3	виконання курсової роботи «Побудова класу Rectangle»	15
	– аналіз поставленої задачі, вибір об'єктів для рішення	3
	– складовий состав об'єктів, назва, методи, що вони повинні виконувати, данні члени	3
	– програмні алгоритми методів та їх відладка	3
	– програмний алгоритм всієї роботи	3
	– оформлення звіту та його захист	3
4	підготовка до екзамену	30

### 9. МЕТОДИ КОНТРОЛЮ

Усний метод, письмовий, тестовий, практична перевірка, а також методи самоконтролю та самооцінки.

### 10. ПОРЯДОК ТА КРИТЕРІЇ ОЦІНЮВАННЯ

Контроль успішності студента здійснюється за допомогою 100-бальної системи оцінювання, що має відповідні оцінки в національній шкалі і шкалі ECTS.

**II семестр****Змістовий модуль 1. Інкапсуляція**

№ п/п	Вид навчальної роботи студента	Максимальна кількість балів
1	Виконання лабораторних робіт: 1. Виконання програм простої структури. Робота з введенням та виведенням даних.	35
	2. Операції та вирази. Операції з цілими числами.	35
2	Контрольна робота	30
<b>Разом</b>		<b>100</b>

**III семестр****Змістовий модуль 2. Поліморфізм та спадкування**

№ п/п	Вид навчальної роботи студента	Максимальна кількість балів
1	Виконання лабораторних робіт: 1. Функції. Передача масивів в функцію та із функції. Робота з динамічною пам'яттю.	35
	2. Робота з двовимірним масивом як з об'єктом. Реалізація суми та різниці об'єктів масивів.	35
2	Контрольна робота	30
<b>Разом</b>		<b>100</b>

**Критерії оцінювання лабораторних робіт**

Передбачено 4 лабораторних роботи. Максимальна кількість балів за роботу - 35 балів.

**Кількість балів «35»** – провів письмовий аналіз поставлених задач, зробив необхідні математичні викладки для складу алгоритму, оформив роботу відповідно до встановлених вимог, при захисті правильно відповідав на питання щодо виконаної роботи.

**Кількість балів «25–34»** – ставиться, якщо студент аналіз задач виконав не повністю, математичні викладки зробив з незначними помилками при виведенні формул або у відповідях на теоретичні питання мали місце помилки, які не впливають в цілому на успішне виконання лабораторної роботи.

**Кількість балів «15–24»** – ставиться, якщо студент аналіз задач виконав не повністю, в представленому алгоритмі математичні вирази обробляються з помилками, слабо орієнтується в порядку виконання програмних виразів.

**Кількість балів «5–14»** – ставиться, якщо студент законспектував тільки завдання і алгоритм роботи, не відповідає на більшість запитань при захисті, а в отриманих відповідях допускає принципові помилки.

**Кількість балів «1–4»** – ставиться, якщо студент законспектував тільки завдання, не відповідає на більшість запитань при захисті, а в отриманих відповідях допускає принципові помилки.

**Критерії оцінювання контрольних робіт**

**Контрольна робота** містить 2 теоретичне питання. Максимальна кількість балів за відповідь на питання не перевершує 15 балів.

-вичерпна відповідь на запитання - 15 балів;

-дана вичерпна відповідь на запитання, але у відповідях є незначні неточності, проте студент показав вміння орієнтуватися при прийнятті рішень, використовуючи теоретичні та практичні знання - 11-14 балів;

- відповідь розкриває суть запитання, але без достатньої повноти та обґрунтування теоретичних і практичних знань, що не знижують кінцевих результатів прийнятих рішень – 7-10 балів;
- відповідь розкриває суть запитання без достатньої повноти та обґрунтування теоретичних і практичних знань – 4-6 балів;
- дана помилкова відповідь на поставлене запитання – 1-3 балів.

### **Критерії оцінювання знань студентів на екзамені з дисципліни «Об'єктно-орієнтоване програмування»**

Екзамен проводиться в письмовій формі у вигляді відповідей на білети, що містять 2 теоретичних питання і 1 практичне завдання (задача). До складання екзамену допускаються студенти, які повністю виконали навчальний план дисципліни – отримали не менше 60 балів з кожного змістового модуля.

Екзамен (максимальна кількість балів – 100):

Максимальна кількість балів за відповідь на кожне теоретичне питання – 30, розв'язання задачі – 40 балів.

– **30 балів** – ставиться за змістовну, логічну, послідовну, правильну відповідь в письмовій формі на питання екзаменаційного білета. При цьому повністю розкриті усі пункти питання.

– **20–29 балів** – ставиться за здебільшого правильну відповідь в письмовій формі на питання екзаменаційного білета, але при порушенні послідовного викладення матеріалу, окремі підпункти питання розкриті не в повному обсязі, у наведених прикладах є незначні помилки синтаксичного або семантичного характеру.

– **11–19 балів** – ставиться за частково правильну відповідь в письмовій формі на питання екзаменаційного білета, якщо вона поверхова, відсутня логічна послідовність відповіді. Наведені приклади свідчать про слабкі знання з теоретичної складової тематики питання.

– **1–10 балів** – ставиться за частково правильну відповідь в письмовій формі на питання екзаменаційного білета, якщо в ній відсутні відповіді на окремі його частини, наявні грубі теоретичні помилки.

За розв'язання задачі ставиться:

**40 балів** – якщо задача розв'язана без помилок і студент дав змістовні письмові пояснення.

**30–39 балів** – якщо задача розв'язана без помилок, але пояснення дано з помилками.

**20–29 балів** – якщо задача розв'язана з синтаксичними помилками, але пояснення свідчать про правильні думки щодо розв'язання задачі.

**10–19 балів** – якщо задача розв'язана не до кінця, але пояснення свідчать про правильні думки щодо розв'язання задачі.

**1–9 балів** – якщо задача розв'язана не до кінця, а думки щодо її розв'язання містять багато помилок.

### **Критерії оцінки курсової роботи з дисципліни «Об'єктно-орієнтоване програмування»**

Максимальна кількість балів за виконання курсової роботи – 100 балів

у т.ч. – виконання курсової роботи – 60 балів;

– захист курсової роботи – 40 балів.

### **Критерії оцінювання виконання та захисту курсової роботи з дисципліни «Об'єктно-орієнтоване програмування»**

Виконання курсової роботи 60 балів.



**1 проміжний контроль, кількість балів – 30.**

№ п/п	Зміст питань	Бали
1.1.	Обґрунтування кількості використовуваних у роботі класів, їх назва та структура	1–10
1.2.	Обґрунтування кількості і типів даних членів кожного класу, їх імена	1–10
1.3.	Назва і описання роботи методів кожного класу	1–10
<b>Всього</b>		<b>30</b>

**2 проміжний контроль, кількість балів – 30.**

№ п/п	Зміст питань	Бали
2.1.	Математичний аналіз роботи кожного методу програмного проекту з наданням, якщо потрібно, математичних викладок та формул	1–5
2.2.	Представлення найважливіших частин алгоритму для кожного методу проекту	1–5
2.3.	Представлення, у вигляді схеми, основних блоків програмного проекту та зв'язків між ними	1–10
2.4.	Роздрукування робочого алгоритму проекту	1–10
<b>Всього</b>		<b>30</b>

**Захист курсової роботи, максимальна кількість балів – 40.****Критерії захисту курсової роботи**

Максимальна кількість балів – 40 балів.

**40 балів** – під час захисту роботи студент вільно орієнтується в алгоритмі свого проекту і безпомилково відповідає на питання щодо його роботи.

**31–39 балів** – під час захисту роботи студент дає вичерпну відповідь на запитання, але іноді плутається у складових частинах представленого алгоритму.

**21–30 балів** – під час захисту роботи студент показує вміння орієнтуватися при прийнятті рішень, використовуючи теоретичні та практичні знання, але допускає помилки, при відповідях плутається в питаннях щодо внесення змін в алгоритм проекту.

**11–20 балів** – під час захисту дає відповіді, що розкривають суть запитання, але без достатньої повноти та обґрунтування теоретичних і практичних знань. Допускає грубі помилки в питаннях щодо пояснень роботи методів класу.

**1–10 балів** – заслуговує студент, більшість пояснень якого не є переконливими та вичерпними, він припустився принципових помилок в теоретичних і практичних питаннях роботи складових частин алгоритму.

**Підсумкова оцінка** з дисципліни в II семестрі визначається за результатами змістового модуля 1.

**Підсумкова оцінка** з дисципліни в III семестрі визначається як середньоарифметична за змістовий модуль 2 та оцінкою екзамену.

**Порядок зарахування пропущених занять** Студенти самостійно вивчають матеріал, готують реферат за темою пропущеної лекції та захищають його у відведений викладачем час. Пропущені практичні і лабораторні заняття відпрацьовуються студентами виконанням відповідної практичної і лабораторної роботи самостійно та її захистом.

## 11. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Основна

1. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. — СПб.: Питер, 2003. — 461 с: ил.
2. Хортон А. Visual C++ 2010: полный курс.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 1216 с.: ил.
3. Информатика: Учеб. пособие. В 2-х кн. – Д.: ПГАСА, 2015.-И74 Кн. 2. Алгоритмизация и программирование / Под. ред. д. т. н., проф. Н.М. Ершовой – Д.: ПГАСА, 2015. – 404 с.

### Допоміжна

1. Коплиен Дж. Программирование на C++. Классика CS. – СПб.: Питер; 2005. – 479 с.: ил.
2. Прата, Стивен. Язык программирования C++. Лекции и упражнения, 6-е изд. : Пер. с англ. — М. : ООО «И.Д. Вильямс», 2012. - 1248 с. : ил.
3. Страуструп Б. Язык программирования C++. Специальное издание. Пер. с англ. — М.: Издательство Бином, 2011 г. — 1136 с: ил.
4. Павловская Т. А., Щупак Ю. А. C++. Объектно-ориентированное программирование: Практикум. — СПб.: Питер, 2016. — 265 с: ил.

## 12. INTERNET-РЕСУРСИ

1. Курс по мові програмування C++ (Електронний ресурс) / Режим доступу: [www.intuit.ru/studies/courses/16740/1301/info](http://www.intuit.ru/studies/courses/16740/1301/info)
2. Курс введення до мови програмування C++ (Електронний ресурс) / Режим доступу: [www.intuit.ru/studies/courses/1039/231/info](http://www.intuit.ru/studies/courses/1039/231/info)
3. <http://www.buildercpp.net.ru/phpBB2/portal.php> Сайт для ІТ програмістів

Розробник \_\_\_\_\_ (Ю. Є. Власенко)  
(підпис)

Гарант освітньої програми \_\_\_\_\_ (Н. М. Єршова)  
(підпис)

Силабус затверджено на засіданні кафедри прикладної математики та інформаційних технологій

Протокол від «12» жовтня 2019 року № 3