

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

Кафедра програмної інженерії та інформаційних технологій управління
(назва)

«ЗАТВЕРДЖУЮ»

Голова науково-методичної комісії _____
(назва комісії)

_____ (підпис) _____ (ініціали та прізвище)

« _____ » _____ 20 _____ року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ (частина 1)

(назва навчальної дисципліни)

рівень вищої освіти перший (бакалаврський)
перший (бакалаврський) / другий (магістерський)

галузь знань 12 Інформаційні технології
(шифр і назва)

спеціальність 126 Інформаційні системи та технології
(шифр і назва)

вид дисципліни професійна підготовка
(загальна підготовка / професійна підготовка)

форма навчання денна
(денна / заочна)

Харків – 2017 рік

ЛИСТ ЗАТВЕРДЖЕННЯ

Робоча програма з навчальної дисципліни

АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ (частина 1)

(назва дисципліни)

Розробники:

доцент, к.т.н., доц.
(посада, науковий ступінь та вчене звання)

(підпис)

Воловщик В.Ю.
(ініціали та прізвище)

(посада, науковий ступінь та вчене звання)

(підпис)

(ініціали та прізвище)

Робоча програма розглянута та затверджена на засіданні кафедри

програмної інженерії та інформаційних технологій управління

(назва кафедри)

Протокол від « 31 » серпня 2017 року № 1

Завідувач кафедри _____ Годлевський М.Д.
(назва кафедри) (підпис) (ініціали та прізвище)

ЛИСТ ПЕРЕЗАТВЕРДЖЕННЯ РОБОЧОЇ НАВЧАЛЬНОЇ ПРОГРАМИ

Дата засідання кафедри – розробника РПНД	Номер протоколу	Підпис завідувача кафедри	Підпис голови НМК (для дисциплін загальної підготовки та дисциплін професійної підготовки за спеціальністю) або завідувача випускової кафедри (для дисциплін професійної підготовки зі спеціалізації, якщо РПНД розроблена не випусковою кафедрою)

МЕТА, КОМПЕТЕНТНОСТІ, РЕЗУЛЬТАТИ НАВЧАННЯ ТА СТРУКТУРНО-ЛОГІЧНА СХЕМА ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета навчальної дисципліни “Алгоритмізація та програмування (частина 1)” є:

- формування у студентів знань за дисципліною, пов’язаних з процесом розробки програм, зокрема включаючи зв’язки між задачею та алгоритмом її вирішення, між алгоритмом та програмним кодом;
- навчання студентів основним положенням мови програмування C++;
- знайомство студентів з інструментальними програмними та технологічними рішеннями, які використовуються при розробці алгоритмічного та програмного забезпечення;
- набуття практичних навичок роботи в колективі, вмінь знаходити правильні рішення для побудови та налагоджування програм мовою C++;
- надання студентам теоретичних основ мови програмування C++ та набуття студентами практичних навичок її використання при розробці програм, заснованих на принципах структурного та базових принципах процедурно-орієнтованого програмування.

Компетентності:

Загальні компетентності:

- Здатність застосовувати знання у практичних ситуаціях.
- Знання та розуміння предметної області та розуміння професійної діяльності.
- Здатність вчитися і оволодівати сучасними знаннями.
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальні (фахові) компетентності:

- Здатність до алгоритмічного та логічного мислення.

Нормативний зміст підготовки здобувачів вищої освіти, сформульований у термінах результатів навчання

Класифікація компетентностей за НРК	Знання	Уміння	Комунікація	Автономія та відповідальність
ФК 9 Здатність до алгоритмічного та логічного мислення.	Знання основних принципів, правил та положень використання базових структурних елементів, що використовуються при побудові	Визначати необхідність використання коректних алгоритмічних інструкцій до вирішення задачі дослідження та їх аналогів. Визначати необхідність	Здатність донесення до фахівців і нефахівців інформації, ідей, проблем, рішень та власного досвіду в галузі професійної діяльності; здатність ефективно	Здатність вивчати нові технології, методи та прийоми щодо розробки алгоритмічного та програмного забезпечення; будувати алгоритмічне та програмне

	алгоритмічно о забезпечення задачі дослідження. Знання основних принципів побудови програмного коду з використанням в якості вихідної інформації алгоритмічно о забезпечення задачі дослідження	використання коректних програмних інструкцій та їх аналогів до побудови програмного коду з використанням в якості вихідної інформації алгоритмічно о забезпечення задачі дослідження	формуванню комунікаційну стратегію.	забезпечення для вирішення задачі дослідження з використанням структурного та базових принципів процедурно-орієнтованого підходів
--	---	--	-------------------------------------	---

Структурно-логічна схема вивчення навчальної дисципліни

Попередні дисципліни:	Наступні дисципліни:
“Інформатика” (шкільна програма)	“Алгоритмізація та програмування (частина 2)”

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

(розподіл навчального часу за семестрами та видами навчальних занять)

Семестр	Загальний обсяг (годин) / кредитів ECTS	З них		За видами аудиторних занять (годин)			Індивідуальні завдання студентів (КП, КР, РГ, Р, РЕ)	Поточний контроль (кількість робіт)	Семестровий контроль	
		Аудиторні заняття (годин)	Самостійна робота (годин)	Лекції	Лабораторні заняття	Практичні заняття, семінари			Залік	Екзамен
1	2	3	4	5	6	7	8	9	10	11
1	180 /6	64	116	32	32			6		1

Співвідношення кількості годин аудиторних занять до загального обсягу складає 35.5%.

СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№ з/п.	Види навчальних занять (Л, ЛР, ПЗ, СР)	Кількість годин	Номер семестру (якщо дисципліна викладається у декількох семестрах). Назви змістових модулів. Найменування тем та питань кожного заняття. Завдання на самостійну роботу.	Рекомендована література (базова, допоміжна)
1	2	3	4	5
Змістовий модуль 1. Основні принципи алгоритмізації та програмування				
	Л ЛР СР	4 4 14.5	Тема 1. Алгоритми. Базові поняття Етапи вирішення задач з використанням обчислювальної техніки. Поняття алгоритму. Діаграма діяльності, як стандартна форма представлення алгоритму. Дані та величини. Обчислювальна техніка в ролі виконувача алгоритмів. Лінійні алгоритми.	[1-5] [6-10] [11-15]
	Л ЛР СР	4 4 14.5	Тема 2. Алгоритми з розгалуженням та циклічними конструкціями Логічні основи алгоритмізації. Алгоритми з розгалуженням. Циклічні конструкції. Структуризація алгоритмів	[1-5] [6-10] [11-15]
	Л ЛР СР	4 4 14.5	Тема 3. Базові поняття щодо мови програмування C++ Типи даних. Змінні, константи. Початкові відомості щодо синтаксису мови програмування C++. Базові операції. Пріоритет операцій. Базові оператори. Потоки вводу та виводу. Керуючі інструкції мови C++. Найпростіша програма на C++	[1-5] [6-10] [11-15]
	Л ЛР СР	4 4 14.5	Тема 4. Управляючі оператори мови C++, що програмують алгоритми з розгалуженням та циклічними конструкціями Оператори безумовного переходу. Умовний оператор. Оператор switch. Оператори циклу з передумовою та постумовою.	[1-5] [6-10] [11-15]
Змістовий модуль 2. Базові принципи процедурно-орієнтоване програмування				
	Л ЛР СР	4 4 14.5	Тема 5. Посилання. Масиви Посилання. Масиви, оголошення ініціалізація та застосування масивів. Багатовимірні масиви.	[1-5] [6-10] [11-15]
	Л ЛР СР	4 4 14.5	Тема 6. Вказівники Оголошення та застосування вказівників. Ініціалізація вказівників. Операції над вказівниками. Вказівники та масиви. Масиви вказівників. Адресна арифметика. Використання динамічної пам'яті	[1-5] [6-10] [11-15]

№ з/п.	Види навчальних занять (Л, ЛЗ, ПЗ, СР)	Кількість годин	<p>Номер семестру (якщо дисципліна викладається у декількох семестрах).</p> <p>Назви змістових модулів.</p> <p>Найменування тем та питань кожного заняття.</p> <p>Завдання на самостійну роботу.</p>	Рекомендована література (базова, допоміжна)
	Л ЛР СР	4 4 14.5	<p>Тема 7. Функції</p> <p>Оголошення та визначення функцій. Змінні, як параметри функцій. Оператор return. Області видимості. Прототип функції. Передача параметрів функції за значенням.</p>	<p>[1-5]</p> <p>[6-10]</p> <p>[11-15]</p>
	Л ЛР СР	4 4 14.5	<p>Тема 8. Рядки</p> <p>Масиви символів. Рядки. Функції роботи з рядками. Області видимості</p>	<p>[1-5]</p> <p>[6-10]</p> <p>[11-15]</p>
Разом		180		
(годин)				

САМОСТІЙНА РОБОТА

№ з/п	Назва видів самостійної роботи	Кількість годин
1	Опрацювання лекційного матеріалу	29
2	Підготовка до практичних(лабораторних, семінарських) занять	29
3	Самостійне вивчення тем та питань, які не викладаються на лекційних заняттях	29
4	Інші види самостійної роботи	29
	Разом	116

ПЕРЕЛІК КОНТРОЛЬНИХ ПИТАНЬ

1. Поняття алгоритму.
2. Способи подання алгоритму.
3. Типи алгоритмів
4. Поняття блок-схеми та діаграми діяльності: нотації, різниця.
5. Константа. Змінна. Вираз. Операції
6. Оператор goto
7. Оператор break
8. Оператор continue
9. Інструкції if, if...else,
10. Інструкція switch
11. Інструкція while
12. Інструкція do...while
13. Інструкція for
14. Посилання
15. Вказівники
16. Взаємозв'язок змінних, посилань та вказівників при вирішенні практичних задач
17. Принципи роботи з динамічною пам'яттю
18. Визначення функції. Прототип функції
19. Види областей видимості та їх особливості
20. Адресна арифметика

МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни для активізації навчального процесу передбачено застосування сучасних навчальних технологій, таких, як: проблемні лекції; робота в малих групах; семінари-дискусії; кейс-метод; ділові ігри.

Проблемні лекції спрямовані на розвиток логічного мислення студентів. Коло питань теми лекції обмежується двома-трьома ключовими моментами, увага студентів концентрується на матеріалі, що не знайшов широкого відображення в підручниках, використовується досвід закордонних навчальних закладів з роздаванням студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються. При викладанні лекційного матеріалу студентам пропонуються питання для самостійного розмірковування. При цьому лектор задає запитання, які спонукають студента шукати розв'язання проблемної ситуації. Така система примушує студентів сконцентруватися і почати активно мислити в пошуках правильної відповіді.

На початку проведення проблемної лекції необхідно чітко сформулювати проблему, яку необхідно вирішити студентам. При викладанні лекційного матеріалу слід уникати прямої відповіді на поставлені запитання, а висвітлювати лекційний матеріал таким чином, щоб отриману інформацію студент міг використовувати при розв'язанні проблеми.

Міні-лекції передбачають викладення навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Міні-лекції проводяться, як правило, як частина заняття-дослідження. На початку проведення міні-лекції за вказаними темами лектор акцентує увагу студентів на необхідності представити викладений лекційний матеріал у так званому структурно-логічному вигляді. На розгляд виносяться питання, які зафіксовані у плані лекцій, але викладаються вони стисло. Лекційне заняття, проведене у такий спосіб, пробуджує у студента активність та увагу при сприйнятті матеріалу, а також спрямовує його на використання системного підходу при відтворенні інформації, яку він одержав від викладача. Проблемні лекції та міні-лекції доцільно поєднувати з такою формою активізації навчального процесу, як робота в малих групах.

Робота в малих групах дає змогу структурувати лекційні або лабораторні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування. Після висвітлення проблеми (при використанні проблемних лекцій) або стислого викладання матеріалу (при

використанні міні-лекцій) студентам пропонується об'єднуватися у групи по 5-6 осіб та презентувати наприкінці заняття своє бачення та сприйняття матеріалу.

Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи, звіту про виконання індивідуальних завдань. Однією з позитивних рис презентації та її переваг при використанні в навчальному процесі є обмін досвідом, який здобули студенти при роботі у певній малій групі.

Лабораторні заняття (з елементами семінарської дискусії) дозволяють формувати у студентів навички практичної реалізації практичних задач шляхом розробки алгоритмічного забезпечення та побудови на основі останнього відповідного програмного забезпечення мовою програмування C++ з використанням структурного та базових принципів процедурно-орієнтованого програмування, узагальнювати отримані результати, формулювати висновки та думки, вести подальший обмін думками та поглядами з іншими учасниками щодо отриманих результатів досліджень з даної проблематики, а також розвивають творче мислення, допомагають формувати погляди і переконання, вчать об'єктивно оцінювати результати і пропозиції опонентів, критично підходити до власних результатів та поглядів.

Ділові та рольові ігри – форма активізації студентів, за якої вони задіяні в процесі інсценізації певної виробничої ситуації у ролі безпосередніх учасників подій. Наприклад, при проведенні лабораторного заняття за темою “Циклічні інструкції мови програмування C++” слід поділити аудиторію на групи, кожній з яких дати певне завдання, реалізація яких повинна бути виконані шляхом використання різних підходів.

Кейс-метод – метод аналізу конкретних ситуацій, який дає змогу наблизити процес навчання до реальної практичної діяльності спеціалістів і передбачає розгляд виробничих, управлінських та інших ситуацій, складних конфліктних випадків, проблемних ситуацій, інцидентів у процесі вивчення навчального матеріалу.

Розподіл форм та методів активізації процесу навчання за темами навчальної дисципліни

Тема	Практичне застосування навчальних технологій
ТЕМА 1. Алгоритми з розгалуженням та циклічними конструкціями	Проблемна лекція “Визначення необхідності та використання алгоритмів з розгалуженням та циклічними конструкціями при вирішенні практичних задач”
ТЕМА 3. Базові поняття щодо мови програмування C++	Міні-лекція “Пріоритети операцій”
ТЕМА 6. Вказівники	Кейс "Робота з пам'яттю" Міні-лекція “Методика роботи з вказівниками на основі використання статичної та динамічної пам'яті”

Тема	Практичне застосування навчальних технологій
ТЕМА 7. Функції	Ділова гра “Обґрунтування особливостей виділення в задачі дослідження підзадач, які можуть бути формалізовані функціями мови програмування C++”

МЕТОДИ КОНТРОЛЮ

Система оцінювання знань, вмінь та навичок студентів передбачає виставлення оцінок за усіма формами проведення занять. Перевірка та оцінювання знань студентів може проводитись у таких формах:

1. Оцінювання роботи студентів у процесі лабораторних занять.
2. Проведення проміжного контролю.
3. Проведення модульного контролю.

Загальна модульна оцінка складається з поточної оцінки, яку студент отримує під час лабораторних занять та оцінки за виконання модульної контрольної роботи.

Загальна оцінка з дисципліни визначається як середнє арифметичне модульних оцінок.

Порядок поточного оцінювання знань студентів

Поточне оцінювання здійснюється під час проведення лабораторних занять і має на меті перевірку рівня підготовленості студента до виконання конкретної роботи. Об'єктами поточного контролю є:

- 1) активність та результативність роботи студента протягом семестру над вивченням програмного матеріалу дисципліни; відвідування занять;
- 2) виконання проміжного контролю;
- 3) виконання модульного контрольного завдання.

Контроль систематичного виконання самостійної роботи та активності на лабораторних заняттях

Оцінювання проводиться за 5-бальною шкалою за такими критеріями:

- 1) розуміння, ступінь засвоєння теорії та методології проблем, що розглядаються;
- 2) ступінь засвоєння матеріалу дисципліни;
- 3) ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються;
- 4) уміння поєднувати теорію з практикою при розгляді виробничих ситуацій, розв'язанні задач, проведенні розрахунків при виконанні завдань, винесених для самостійного опрацювання, та завдань, винесених на розгляд в аудиторії;

5) логіка, структура, стиль викладу матеріалу в письмових роботах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Оцінка "відмінно" ставиться за умови відповідності виконаного завдання студента або його усної відповіді до всіх п'яти зазначених критеріїв.

Відсутність тієї чи іншої складової знижує оцінку на відповідну кількість балів.

При оцінюванні практичних завдань увага приділяється також їх якості та самостійності, своєчасності здачі виконаних завдань викладачу (згідно з графіком навчального процесу). Якщо якась із вимог не буде виконана, то оцінка буде знижена.

Проміжний модульний контроль

Проміжний модульний контроль рівня знань передбачає виявлення опанування студентом матеріалу лекційного модуля та вміння застосовувати його для вирішення практичної ситуації і проводиться у вигляді контрольної роботи за темами 1-го або 2-го модулю.

Проведення модульного контролю

Модульний контроль здійснюється та оцінюється за допомогою проведення контрольної роботи за всіма темами дисципліни.

Підсумкова оцінка з дисципліни розраховується як середня з кількох складових, що враховує оцінки кожного виду контролю (дві оцінки за результатами поточного модульного контролю і підсумкову контрольну роботу).

РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ, ТА ШКАЛА ОЦІНЮВАННЯ ЗНАТЬ ТА УМІНЬ (НАЦІОНАЛЬНА ТА ECTS)

Таблиця 1 – Розподіл балів

	Поточний контроль		Семестровий контроль	Всього за семестр
	Контр Р	лр		
Підсумкові бали	75		25	100
Макс. проміжні бали	20	15		
Кільк. од. обліку у семестрі	6	6		
Макс. проміжних балів, всього	32	43		100
Коеф.. перерахунку	1			
Макс. кільк. підсумкових балів	32	43	25	100

Таблиця 2 – Шкала оцінювання знань та умінь: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 ... 100	A	відмінно
82 ... 89	B	добре
74 ... 81	C	
64 ... 73	D	задовільно
60 ... 63	E	
35 ... 59	FX	незадовільно з можливістю повторного складання
0 ... 34	F	незадовільно з обов'язковим повторним вивченням дисципліни

НАВЧАЛЬНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Воловщиків В.Ю. Лабораторний практикум з дисципліни “Алгоритмізація та програмування (частина 1)” [Електронний ресурс]. – Режим доступу: ntumoodle.com

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базова

1. Буч Г., Рамбо Дж. Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. - М.: ДМК, 2000.
2. Дейтел Х., Дейтел П. Как программировать на C++: Третье издание. - М.: ЗАО "Издательство БИНОМ", 2008. – 1454с.
3. Ковалюк Т.В. Основи програмування. – К.: Питер, 2006.
4. Ставровський А.Б., Карнаух Т.О. Програмування. Перші кроки. – К.: Діалектика, 2004.
5. Страуструп Б. Язык программирования C++. Третье издание – СПб. :М.: Бином, 2004. – 203с.

Допоміжна література

6. Шилдт Г. С++ руководство для начинающих. – М.-СПб.-К.: Вильямс, 2005. – 672с.
7. Шилдт Г. Самоучитель С++. – СПб.: БХВ-Петербург, 2003. – 688с.
8. Марченко А.Л. С++ бархатный путь. – М.: Горячая линия-телеком, 2005. – 399с.
9. Дэвис Стефан Р. С++ для “чайников”. – М.-СПб.-К.: Диалектика, 2003. – 336с.
10. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования. – М.: Форум, 2008. – 432с.

ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

- 11 <http://www2.research.att.com/~bs/C++.html>
- 12 <http://cs.nyu.edu/courses/summer12/CSCI-GA.2110-001/downloads/C++%20Standard%202003.pdf>
- 13 <http://www.cplusplus.com/>
- 14 <http://www.learncpp.com/>
- 15 <http://uk.wikibooks.org/wiki/C++>