

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ

КАФЕДРА: “ПРОГРАМНОЇ ІНЖЕНЕРІЇ”

МЕТОДИЧНІ ВКАЗІВКИ
щодо самостійної роботи студентів
та модульного контролю знань
з дисципліни

“ОСНОВИ ПРОГРАМУВАННЯ”

для студентів напрямку підготовки
6.050103 – Програмна інженерія

ТЕРНОПІЛЬ 2015

Методичні вказівки щодо самостійної роботи студентів та модульного контролю знань з дисципліни “Основи програмування” для студентів напряму підготовки 6.050103 – Програмна інженерія. / Уклад.: М.Петрик, О.Петрик - Тернопіль: ТНТУ 2015 - 22 с.

Призначені для полегшення засвоєння дисципліни “Основи програмування” і контролю знань студентів. Складається з урахуванням модульної системи навчання, рекомендацій до самостійної роботи і індивідуальних завдань, тем практичних та лабораторних занять, тестів, екзаменаційних питань, типової форми та вимог для комплексної перевірки знань з дисципліни

Укладачі: О.Петрик, М.Петрик

Розглянуто на засіданні кафедри програмної інженерії,
протокол № 3 від 16.10.2015р.

Схвалено на засіданні методичної ради факультету комп’ютерно-інформаційних систем і програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя, протокол №3 від 22 жовтня 2015р.

ВСТУП

Щоб навчити студентів основам алгоритмізації, необхідне, насамперед, прагнення самого студента досягти успіху). Навчити програмувати можна тільки студента, який сам хоче навчитись.

Мета. Навчити студентів основам алгоритмізації, особливостям програмування мовами високого рівня, основним прийомам програмування з урахуванням сучасних концепцій і тенденцій розвитку технологій програмування, розв'язувати за допомогою ЕОМ реальні науково-технічні задачі різної складності, сформулювати у слухачів погляд на програмування, як на цілісне поєднання науки, техніки та мистецтва, пов'язане з математикою, логікою, психологією та іншими галузями знань.

Дисципліна знайомить з поняттями алгоритму і його програмної реалізації, основними керуючими конструкціями і типами даних, дає базові знання однієї з мов програмування, вчить працювати з функціями, масивами, файлами. Вивчаються базові структури даних, прості алгоритми пошуку і сортування, загальні питання проектування, написання та налагодження програм. Курс також містить введення в історичний та соціальний контекст програмування, дає огляд операційних систем та мережі Інтернет.

Від студентів не вимагається попереднього досвіду в галузі програмування та інформатики.

Завдання.

У результаті вивчення навчальної дисципліни студент повинен

знати:

що таке мова програмування, типи мов програмування та їх особливості, середовище програмування; що таке дані, тип даних, класифікацію типів даних, базові структури даних; основні алгоритмічні конструкції та їх реалізацію мовою програмування C/C++; прості алгоритми пошуку і сортування; загальні принципи проектування, написання та налагодження програм в процедурно-орієнтованому стилі.

вміти:

створювати невеликі програми в структурній парадигмі; реалізувати усі етапи розв'язування задачі: від постановки до отримання кінцевого результату; використовувати функції при написанні програм; реалізувати багатомодульні програми; застосовувати алгоритми пошуку та сортування масивів, основні кроки для

роботи з файлами; користуватися стандартною бібліотекою мови, можливостями середовища програмування при написанні та налагодженні програм.

У процесі вивчення дисципліни студенти повинні набути навички проектування алгоритму розв'язку поставленої задачі, зображення алгоритму у вигляді блок-схем, реалізації алгоритму мовою програмування, тестування і налагодження програмної системи, застосування основних алгоритмічних конструкцій, парадигм модульного і структурного програмування.

Завданням лекційних занять є ознайомлення студентів з основними принципами програмування, базовими елементами мови програмування C/C++, засобами їх реалізації, особливостями використання операторів мови.

Завдання лабораторних занять полягає у вивченні студентами основних прийомів програмування.

Знання, вміння та навички, отримані студентами під час вивчення даної дисципліни будуть необхідними для їхньої професійної діяльності, а також використовуватимуться при написанні курсових та дипломних проектів з програмної інженерії.

У цих методичних вказівках для полегшення засвоєння матеріалу приведені контрольні питання для самоперевірки.

1. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ І ТЕМИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

1.1 Структура навчальної дисципліни за модульною системою

Для вивчення навчальної дисципліни “Основи програмування” відводиться:
288 год (8 кредитів) для денної форми навчання і 270 год (7,5 кредитів) – для заочної.

Робоча програма з дисципліни містить 2 змістових модулів.

Погодинний розподіл тем курсу наведено у таблиці 1

Таблиця 1 - Погодинний розподіл тем курсу

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					Усього	у тому числі				
л		п	лаб	інд	с.р.	л		п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13
Модуль 1												
Змістовий модуль 1. Організація програм												
Тема 1. Вступ до дисципліни. Основні поняття та визначення	14	4	–	2	–	8	14	–	–	–	–	14
Тема 2. Елементи мови	16	4	–	4	–	8	15	1	–	–	–	14
Тема 3. Організація введення-виведення в С/С++	18	4		4		10	15	1	–	–	–	14
Тема 4. Базові структури алгоритмів і їх реалізація засобами мови С/С++	22	4	–	6	–	12	18	2	–	2	–	14
Тема 5. Процедурно-орієнтоване програмування. Введення у функції	20	4	–	4	–	12	15	1	–	–	–	14
Тема 6. Функції	18	4		4	–	10	17	1	–	2	–	14
Тема 7. Вказівники. Посилання	20	4	–	4	–	12	18	2	–	2	–	14
Тема 8. Методології розробки програм, поняття модульного програмування	18	4	–	4	–	10	15	1	–	–	–	14
Тема 9. Директиви препроцесора	18	4		4	–	10	15	1	–	–	–	14
Разом за змістовим модулем 1	164	36	–	36	–	92	142	10	–	6	–	126

Таблиця 1 – Продовження. Погодинний розподіл тем курсу

1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 2. Структури даних і алгоритми												
Тема 1. Організація даних та алгоритми їх обробки. Масиви	24	6	–	6	–	12	24	2	–	4	–	18
Тема 2. Символьні масиви	18	4	–	4	–	10	22	2	–	2	–	18
Тема 3. Типи даних користувача	24	6	–	6	–	12	24	2	–	4	–	18
Тема 4. Файлові структури даних	20	4	–	4	–	12	22	2	–	2	–	18
Тема 5. Способи організації доступу до файлів	20	4	–	4	–	12	20	–	–	2	–	18
Тема 6. Алгоритмізація типових обчислювальних задач	18	4	–	4	–	10	22	–	–	4	–	18
Разом за змістовим модулем 2	124	28	–	28	–	68	120	–	–	18	–	108
Усього годин	288	64	–	64	–	160	270	8	–	28	–	234

1.2 Контрольні питання для самоперевірки

Питання для самоперевірки складені за матеріалами всієї дисципліни “Основи програмування” і є для студентів допоміжним засобом вивчення пропонованого курсу. Нижче приводяться складені питання щодо дисципліни.

1. Етапи розв’язування задач на ЕОМ.
2. Оперативна пам’ять процесора. Регістри процесора. Представлення інформації в пам’яті ПК: представлення чисел з фіксованою і плаваючою точкою.
3. Історія виникнення мови C/C++.
4. Елементи мови C: алфавіт, ідентифікатори, константи, коментарі.
5. Структура програми мовою C/C++; поняття препроцесора, функції.
6. Загальний формат оголошення типів даних, ініціалізація даних; область видимості.
7. Область дії змінних. Класи пам’яті: auto, extern, register, static.
8. Функція scanf, загальний формат, форматування введення. Приклад.
9. Функція printf, загальний формат, форматування виведення. Приклад.
10. Функції виведення символів і рядків: загальний формат, особливості застосування.

11. Функції введення символів і рядків: загальний формат, особливості застосування.
12. Реалізація потокового введення-виведення.
13. Основні операції: арифметичні операції, операції присвоєння, операції порівняння. Пріоритет операцій.
14. Логічні операції, операція слідування (кома), умовна операція `?:`, операція `sizeof()`. Їх призначення, загальний формат, дія, приклад.
15. Вирази і їх запис мовою C/C++. Приклади.
16. Оператори: порожній оператор і блок.
17. Умовний оператор, умовний вираз. Загальний формат, дія оператора, приклад використання.
18. Оператор переключення. Загальний формат, дія оператора, особливості застосування. Приклад.
19. Оператори циклу, їх види, особливості застосування.
20. Оператор циклу з лічильником. Загальний формат, дія оператора, приклад використання.
21. Оператор циклу з передумовою. Загальний формат, дія оператора, приклад використання.
22. Оператор циклу з післяумовою. Загальний формат, дія оператора, приклад використання.
23. Оператор повернення значень.
24. Поняття перевантаження функцій. Способи перевантаження функцій. Приклади.
25. Вбудовані функції. Загальний формат. Приклад.
26. Функції із змінним числом параметрів. Загальний формат, приклад використання.
27. Функції користувача: оголошення і визначення функції. Тип `void`. Виклик функції.
28. Функції, що не повертають значення. Передача параметрів у функцію.
29. Основні відомості про вказівники: поняття вказівника, оголошення, приклад.
30. Базові типи даних.
31. Операції з вказівниками, приклади.
32. Арифметика з вказівниками. Проблеми, пов'язані з вказівниками.
33. Використання вказівників у функціях в якості параметрів і результату.
34. Поняття даних типу посилання. Основні відмінності вказівників і посилань.

35. Порівняльна характеристика вказівників і посилань.
36. Параметри-посилання. Функції, що повертають значення типу посилання.
37. Поняття складених типів даних і їх класифікація.
38. Поняття масиву, властивості масивів. Розміщення масивів у пам'яті комп'ютера.
Назва масиву, індексація масиву, ідентифікатори і значення елементів масиву.
39. Оголошення, способи ініціалізації цифрових масивів.
40. Доступ до елементів масивів. Вказівники масивів. Доступ до елемента масиву з допомогою індексації вказівника, зміщення вказівника.
41. Масиви як параметри функцій. Масиви вказівників.
42. Динамічний масив, оператори виділення і вивільнення пам'яті під масив.
43. Типові задачі на роботу з масивами.
44. Робота з одновимірними масивами.
45. Операції з матрицями: обчислення суми, добутку, максимуму, мінімуму кожного стовпця (рядка), виділення головної і бічної діагоналей, перестановка заданих рядків (стовпців).
46. Використання масивів у функціях (як параметр і як тип функції).
47. Оголошення, способи ініціалізації рядків. Вбудовані функції обробки рядків: функції введення, виведення рядків, визначення довжини рядка, порівняння, копіювання, конкатенації рядків, розбиття рядка на лексеми, пошук символу, підрядка у рядку . Передача рядків у функції. Рядок як результат виконання функції.
48. Структури, оголошення і ініціалізація. Елемент структури, змінна типу структури. Вказівники на структуру. Доступ до елементів структури. Операції над структурами. Копіювання структури, динамічне виділення пам'яті під структуру. Вкладені структури, масиви структур.
49. Структури як параметри функцій. Повернення значень типу структура.
50. Перейменування типів. Типи, що задаються переліком.
51. Об'єднання. Бітові поля.
52. Файли: структура логічного і фізичного файлів, класифікація файлів, способи організації доступу до файлів.
53. Оголошення даних файлового типу. Функції відкриття і закриття файлу. Функції запису до файлу; створення нового файлу; функції читання з файлу.
54. Функції для реалізації прямого доступу до файлів.
55. Обробка текстових і бінарних файлів.

56. Поняття директив препроцесора. Директиви включення.
57. Директиви макровизначення. Директиви умовної компіляції.
58. Оператор `defined`. Директива `#ERROR`. Директива `#pragma`.
59. Алгоритмізація типових обчислювальних задач
60. Алгоритми сортування. Алгоритми пошуку. Теорія чисел: прості числа: пошук і підрахунок, подільність чисел, арифметика остач. Комбінаторика.

1.3 ТЕМАТИКА ЛАБОРАТОРНИХ ЗАНЯТЬ

Ціль проведення лабораторних занять - закріплення теоретичних положень, викладених на лекціях. Студентам пропонуються лабораторні завдання, проблеми їх вирішення, довідкова література і методичні вказівки, розроблені для самостійної роботи. Викладач контролює хід рішення практичних задач і оцінює уміння студентів застосовувати знання, одержані на лекціях.

Курс об'єднує в собі три складові навчання: теорію, практику і контроль засвоєння матеріалу. Курс наповнений стислими теоретичними відомостями, систематизованими завданнями до лабораторних робіт, перевіреними прикладами програмної реалізації та наборами контрольних тестів

Кожна лабораторна робота містить індивідуальні завдання з теми. В одному варіанті кожної лабораторної роботи є від 1 до 4 завдань. Кількість варіантів в кожній лабораторній роботі відповідає кількості студентів у групі. Завдання до лабораторних робіт підбрані індивідуально, за кількістю студентів, систематизовані за рівнями складності і розподіляються між студентами виходячи з їхнього початкового рівня підготовки в галузі програмування, реалізація яких, надіюсь, сприяє формуванню логічного мислення, та культури програмування, формує творчий стиль програміста та закладає наукові основи оволодіння сучасними технологіями програмування.

Зміст лабораторних занять наведено в табл. 2.

Таблиця 2 – Тематика лабораторних робіт з курсу

№ ЛР	Тема лабораторної роботи	Кількість годин	
		Денна форма	Заочна форма
1	Ознайомлення з середовищем програмування MS Visual C++ (Borland C++): створення консольного додатку, компіляція, виконання, налагодження програми	4	-
2	Створення найпростіших діалогових програм. Реалізація лінійного алгоритму	4+4	2
3	Умовні оператори	4	
4	Цикли	4	
5	Застосування динамічних змінних до рішення задач	4	2

Таблиця 2 – Продовження. Тематика лабораторних робіт з курсу

1	2	3	4
6	Реалізація функцій користувача засобами мови C/C++ Перевантажені, вбудовані функції. Рекурсія	4+4+4	2
7	Одновимірні, двовимірні масиви	6	4
8	Стрічки	4	2
9	Структури та об'єднання	4	4
10	Файли	6	2
11	Реалізація типових обчислювальних задач	6	4
	Разом	64	18

1.4. Теми для самостійного вивчення

Самостійна робота студента є основним видом засвоєння навчального матеріалу у вільний від аудиторних занять час.

Метою самостійної роботи є вироблення студентами навичок і вміння працювати з літературою, віднаходити головні аспекти проблем, що потребують стійкого засвоєння.

Предметом самостійної роботи студентів є опрацювання ними як окремих тем програми курсу в цілому, так і деяких розділів тем, самостійне розв'язування конкретних задач з метою практичного поглиблення здобутих знань.

Перевірка рівня засвоєння матеріалу самостійно опрацьованих тем здійснюється при проведенні поточного тестового контролю з окремих тем згідно програми та під час написання домашніх і лабораторних робіт.

Таблиця 3 – Тематика самостійної роботи з курсу

№ з/п	Назва теми	Кількість Годин	
		денна форма	заочна форма
1	2	3	4
1	Історія мови C. Стандартизація мови C. Консольні додатки. Використання IDE: опції панелі інструментів, проекти і рішення. Налаштування опцій середовища.	8	12
2	Порозрядні логічні операції. Операції зсуву вліво і вправо. Пріоритет і порядок виконання операцій. Оператори порівняння.	8	20

Таблиця 3 – Продовження. Тематика самостійної роботи з курсу

1	2	3	4
3	Розв'язування індивідуальних завдань лабораторних робіт.	30	30
4	Оформлення звітів, підготовка лабораторних робіт до захисту.	20	20
5	Функції стандартної бібліотеки. Генерація випадкових чисел. Класи пам'яті.	8	14
6	Рекурсія в порівнянні з ітерацією. Непряма рекурсія. Рекурсія і час життя даних.	6	14
7	Оголошення вказівників на функції. Використання модифікатора const з вказівниками. Порівняльний аналіз вказівників і посилань.	10	16
8	Параметри функції main.	6	6
9	Директива #pragma. Попередньо визначені макроси. Моделі пам'яті для роботи з вказівниками, дальні та ближні вказівники.	6	8
10	Динамічне виділення пам'яті. Функції динамічного розподілу пам'яті malloc, calloc, free, оператори new і delete	10	16
11	Деякі функції бібліотеки <string>. Пошук символів. Пошук підрядків. Інверсія рядків.	8	12
12	Вказівники на структури. Об'єднання і операції з ними.	10	16
13	Функції позиціювання у файлі.	10	16
14	Основи аналізу алгоритмів. Масиви змінного розміру.	10	17
15	Функції дати, часу і локалізації. Робота з датами, операції з бітами і байтами (підрахунок, зсув, дзеркальне відображення бітів). Бітові поля.	10	17
	Разом	160	234

2. КОНТРОЛЬ ЗНАНЬ СТУДЕНТІВ

Контроль знань студентів включає в себе – виконання індивідуальних завдань, оформлення звіту з лабораторних робіт, захист звітів із лабораторних робіт, поточне комп'ютерне тестування з кожного модуля, підсумковий модульний контроль з дисципліни. Підсумковий модульний контроль передбачає комп'ютерне тестування знань теоретичних питань з дисципліни і рішення практичного завдання згідно варіанту.

Типові тести для контролю знань

1. У фрагмент програми вписати логічний вираз, який визначає кількість додатних елементів у масиві A[n]:

```
k=0;
for ( int i=0; i<n; i++)
    if ( _____ ) k++;
```

2. Задано вектор X[n]. Наступний фрагмент програми визначає:

```
s=0;
for(int i=0;i<n;i++)
    if(X[i]>0)s+=X[i];
```

- а) максимальний елемент масиву;
- б) суму додатних елементів;
- в) кількість додатних елементів;
- г) індекс останнього додатного елемента;
- д) індекс першого додатного елемента.

3. Задано матрицю a[n][n]. Фрагмент програми

```
for ( int i=0;i<n;i++) {
    int c=a[i][i];
    a[i][i]=a[i][n-i-1];
    a[i][n-i-1]=c;
};
```

- а) змінює порядок рядків;
- б) змінює порядок стовпців;
- в) міняє місцями елементи головної та бічної діагоналей;
- г) міняє місцями елементи і-того рядка та j-го стовпця;
- д) нічого не змінює в таблиці.

4. Встановити відповідність між завданням і його реалізацією на C++

- | | |
|---|---|
| а. Оголошення масиву з іменем numbers типу float з 10 елементів і ініціалізація масиву, якщо size=10. | а. float *nPtr; |
| | б. float numbers[size]={0.0, 1.1, 2.2, 3.3, 4.4}; |
| | в. nPtr=numbers; |

- | | |
|--|---|
| <p>b. Оголошення вказівника nPtr на об'єкт типу float</p> <p>c. Присвоєння початкової адреси масиву numbers змінній вказівнику nPtr</p> <p>d. Друк елементів масиву numbers з використанням вказівника nPtr</p> <p>e. Друк елементів масиву numbers з використанням індексації вказівника nPtr</p> | <p>d. <code>for (int i=0; i<size; i+=) cout <<*(nPtr+i)<<' '</code> ;</p> <p>e. <code>for (int i=0; i<size; i+=) cout <<nPtr[i]<<' '</code> ;</p> <p>f. <code>for (int i=0; i<size; i+=) cout <<*(numbers+i) <<' '</code> ;</p> |
|--|---|

5. Структуру test, яка містить поля a - 1 біт, b-2 біти, c- 3 біти, d - 4 біти, e- 6 біт необхідно описати наступним чином:

- a. `struct test {unsigned a/1; unsigned b/2; unsigned c/3; unsigned d/4; unsigned e/6;};`
- b. `struct test {unsigned a*1; unsigned b*2; unsigned c*3; unsigned d*4; unsigned e*6;};`
- c. `struct test{ unsigned a:1; unsigned b:2; unsigned c:3; unsigned d:4; unsigned e:6;};`

6. Файли призначені для

- a. постійного зберігання великих об'ємів даних на зовнішніх носіях
- b. постійного зберігання великих об'ємів даних в оперативній пам'яті
- c. тимчасового зберігання великих об'ємів даних в оперативній пам'яті

7. Режим "r", заданий у функції fopen, призначений для:

- a. відкриття файлу без дозволу на модифікацію, файл відкривається лише для читання
- b. створення нового файлу тільки для запису, якщо файл із вказаним ім'ям вже існує, то він перезапишеться
- c. відкриття файлу для читання та запису

3. ЕКЗАМЕНАЦІЙНІ ПИТАННЯ

Екзаменаційні білети з дисципліни “Основи програмування” складені за всім курсом із внесенням питань, що представлені в розділі 1.2 “Контрольні питання для самоперевірки”.

Кількість білетів - 30 шт. Білети включають всі розділи курсу та є практично рівноцінними за своє складністю.

Зразок екзаменаційного білету:

Тернопільський національний технічний університет імені Івана Пулюя

Кафедра програмної інженерії

Напрямок підготовки 6.050103 “Програмна інженерія”

Дисципліна: Основи програмування

Білет № 10

1. Форматований вивід printf
2. Рекурсивні функції
3. Задача

Затверджено на засіданні кафедри програмної інженерії,
протокол №1 від 31. 08. 2015 р.

Зав. кафедрою програмної інженерії

Екзаменатор

д.ф.н, Петрик М.Р.

Петрик О.Ю.

4. ПАКЕТ КОНТРОЛЬНИХ РОБІТ ДЛЯ КОМПЛЕКСНОЇ ПЕРЕВІРКИ ЗНАНЬ СТУДЕНТІВ З ДИСЦИПЛІНИ

Для контролю засвоєння матеріалу дисципліни розроблений пакет контрольних робіт для комплексної перевірки знань студентів. Дані контрольні роботи дозволяють перевірити рівень “залишкових” знань.

Всі завдання ККР з дисципліни “ Основи програмування” складені за всім курсом із використанням питань, що представлені в розділі 1.2 “Контрольні питання для самоперевірки”.

Кількість завдань - 30 шт. Завдання розроблені так, щоб включали всі розділи курсу.

4.1 Критерії оцінювання

При перевірці ККР за відповіді на кожне питання варіанту виставляється диференційована оцінка.

“Відмінно” - виставляється, якщо при відповіді на питання студент виявив всебічні знання програмного матеріалу, уміння вільно-виконувати завдання, передбачені програмою на рівні творчого використання:

“Добре” - виставляється, якщо при відповіді на питання студент виявив повне? знання програмного матеріалу, уміння успішно виконувати завдання, передбачені програмою на рівні аналогічного відтворення;

“Задовільно” - виставляється, якщо при відповіді на питання студент виявив повні знання основного матеріалу, в об'єкті, що необхідний для подальшого навчання і роботи, здатність упоратися з виконанням завдань, передбачених програмою на рівні репродуктивного відтворення;

“Незадовільно” - виставляється, якщо при відповіді на питання студент виявив серйозні пробіли в знаннях основного програмного матеріалу, допустив принципові помилки при виконанні завдання на рівні нижче репродуктивного відтворення.

Оцінка виконаної ККР визначається за допомогою формули:

$$A = \alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3$$

де $\alpha_1=0.333$, $\alpha_2=0.333$, $\alpha_3=0.333$ коефіцієнти складності завдання;

$\beta_1, \beta_2, \beta_3$ - оцінка за кожне завдання за п'ятибальною системою.

Оцінка "відмінно" ставиться, якщо $A > 4.6$.

Оцінка "добре" ставиться, якщо $3.6 < A < 4.6$.

Оцінка "задовільно" ставиться, якщо $2.6 < A < 3.6$.

Оцінка "незадовільно" ставиться, якщо $A < 2.6$.

5. КОНТРОЛЬНІ ПИТАННЯ З ДИСЦИПЛІНИ

Модуль 1

1. Історія виникнення мови C/C++. Її переваги та недоліки
2. Структура програми на мові C
3. Алфавіт мови C. Лексеми
4. Типи даних в мові C
5. Змінні. Оголошення, область видимості
6. Константи. Оголошення на застосування
7. Вирази і операції.
8. Види операцій
9. Пріоритети операцій і порядок виконання обчислень
10. Форматований вивід *printf*
11. Форматований ввід *scanf*
12. Оператори мови. Види операторів
13. Оператори-вирази
14. Оператор умови *if*
15. Оператор умови *switch*
16. Оператор циклу *for*
17. Оператор циклу *while*
18. Оператор циклу *do-while*
19. Оператори переходу *goto* і *break*
20. Оператори переходу *continue*, *return*.
21. Вказівник. Оголошення та типізація вказівників.
22. Вказівник. Операції з вказівниками
23. Вказівник. Адресна арифметика
24. Посилання. Їх переваги та обмеження
25. Способи використання посилань
26. Функції в мові C. Їх структура
27. Виклик функції. Прототипи функції
28. Взаємодія фактичних і формальних параметрів функції
29. Параметри функцій. Передача масиву
30. Параметри функцій. Передача символьного рядка
31. Параметри функцій. Передача багатовимірного масиву
32. Параметри функцій. Передача структури.
33. Параметри командного рядка

34. Вказівник на функцію. Оголошення та використання вказівника на функцію.
35. Вказівник на функцію як параметр функції
36. Вказівник на функцію як тип значення функції
37. Масиви вказівників на функцію
38. *inline*-функції
39. Рекурсивні функції
40. Функції з неоголошеними параметрами
41. Передача параметрів функції *main*
42. Заголовкові файли

Модуль 2

1. Структури. Звернення до полів структури
2. Об'єднання. Бітові поля
3. Оператор перейменування типу.
4. Масиви в мові C. Оголошення та ініціалізація масивів
5. Масиви в мові C. Способи звернення до елементів масиву
6. Багатовимірні масиви. Звернення до елементів масиву
7. Багатовимірні масиви. Пониження вимірності масиву
8. Символьні рядки. Оголошення та ініціалізація
9. Введення виведення символьних рядків
10. Функції для роботи із символьними рядками
11. Структури. Оголошення та ініціалізація
12. Рівні введення/виведення даних. Файли і потоки
13. Представлення файлу в мові C
14. Функції для роботи із потоками. Відкриття і закриття потоків
15. Функції для роботи із потоками. Стандартні потоки, їх перескерування.
16. Функції для роботи із потоками. Потокове введення/виведення даних
17. Встановлення позиції в файлі
18. Функції аналізу помилок при роботі із файлами
19. Буферизація при потоковому введенні/виведенні. Функції керування буферизацією
20. Поняття динамічної пам'яті.
21. Динамічне виділення пам'яті
22. Зміна розміру та вивільнення динамічно виділеної пам'яті
23. Виняткові ситуації, що виникають під час виконання програми
24. Методи обробки виняткових ситуацій

6. НАВЧАЛЬНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ

Базова

1. Шпак З.Я Програмування мовою С. – Львів: Оріяна-Нова, 2006.-432с.
2. Шилдт Г. С++: базовый курс, 3-е изд. : Пер. с. англ. – М.:Изд. дом «Вильямс», 2010. – 624с.
3. Страуструп Б. Дизайн и эволюция С++: Пер. с англ. - М.:ДМК Пресс; СПб: Питер, 2006 - 448 с.
4. Айвор Хортон. Visual С++ 2010: полный курс. Изд-во: Диалектика-Вильямс, 2011. – 1216 с.
5. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования С++: Учебный курс.- Харьков: Фолио; М.:ООО "Издательство АСТ", 2001.-500 с.
6. Дейтел Х., Дейтел П. Как программировать на С++. 5-е изд. - М.:ООО „Бином-Пресс”, 2008г.–1456с
7. Романов Е.Л. Практикум по программированию на С++: Учебное пособие. СПб: БХВ-Петербург, Новосибирск: Изд-во НГТУ, 2004. – 432 с.
8. Семакин И.Г., Шестаков А.П. Основы программирования: Учебник. - М.: Мастерство,2002.- 432с.

Допоміжна

9. Златопольский Д.М. Сборник задач по программированию. - 2-е изд. СПб.: БХВ - Петербург, 2007. - 240 с.
10. Караванова Т.П. Информатика: основы алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами: Навч. посіб. для 8-9 кл. із поглибл. вивч. інформатики – К.: Генеза. – 2006.- 286 с.
11. Кнут Д. Искусство программирования. т.2. Получисленные алгоритмы.- М., СПб., К.:Вильямс, 2000.- 832 с.
12. Прата С. Язык программирования С++. Лекции и упражнения. 5-е изд. –М.: ООО «И. Д. Вильямс», 2007 – 1184с.
13. Шилдт Г. Полный справочник по С++. 4-е издание. - М.: Изд.дом «Вильямс», 2006г.–800с.
14. Шилдт Г. С++: руководство для начинающих, 2-е изд.: Пер. с. англ. – М.:Изд.дом «Вильямс», 2005. – 672с.

Інформаційні ресурси

1. <http://www.rsa.com>
2. <http://www.thesecuritynetwork.org>
3. <http://www-01.ibm.com/software/tivoli/products/security-network-intrusion-prevention/>
4. <http://www.support.kaspersky.ru/9464>

НАВЧАЛЬНЕ ВИДАННЯ МЕТОДИЧНІ ВКАЗІВКИ ЩОДО САМОСТІЙНОЇ
РОБОТИ СТУДЕНТІВ ТА МОДУЛЬНОГО КОНТРОЛЮ ЗНАНЬ

з дисципліни “Основи програмування” для студентів напряму підготовки 6.050103 –
Програмна інженерія.

Укладачі: М.Петрик, О.Петрик