УНИВЕРСИТЕТ ИМЕНИ СУЛЕЙМАНА ДЕМИРЕЛЯ

Жапаров М. К.

ОСНОВЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ С++

Учебно-методическое пособие

Каскелен, 2015

УДК 004. 4(075.8)

ББК 32.973.26-018 я73

Ж 33

Рекомендовано к изданию Учебно-методическим советом Университета имени Сулеймана Демиреля протокол №1 от 25.08.2015 г.

Рецензенты:

Кожахметов К., PhD, ассистент профессора, Казахстанско-Британский технический университет

Амиргалиев Е. Н. д.т.н., профессор, Университет имени Сулеймана Демиреля

Жапаров М. К.

Ж 33 Основы языка программирования C++: учебно-методическое пособие. Жапаров М. К. — Каскелен, Университет имени Сулеймана Демиреля, 2015. — 86 с. — на русском языке.

ISBN 978-601-7537-22-7

Учебно-методическая разработка «Microlearning in Programming Language C++:» предназначена для бакалавров по специальностям 5B03000 — «Информационные Системы» и 5B040000 — «Вычислительная Техника и Программное Обеспечение».

Methodical-educational manual « Microlearning in Programming Language C++» is designed for bachelors of specialties 5B030000 — “Information Systems” and 5B040000 — “Computing Systems and Software”.

УДК 004. 4(075.8)

ББК 32.973.26-018 я73

© Жапаров М. К.

© Университет имени Сулеймана Демиреля, 2015

ISBN 978-601-7537-22-7

СОДЕРЖАНИЕ

1.1 простая программа в C++ 5
1.2 компилирование и запуск 6
1.3 комментарии 7
1.4 ввод/вывод 8
1.5 типы данных 9
1.6 тип данных integer 10
1.7 тип данных real number 11
1.8 тип данных bool 12
1.9 тип данных character 13
1.10 кастинг типов данных 14
1.11 среда переменных 15
1.12 константы 16
1.13 ключевое слово define 17
1.14 ключевое слово enum 18
2. ИСПОЛЬЗОВАНИЕ ОПЕРАТОРОВ
2.1 арифметические операторы 19
2.краткое написание операторов 20
2.3 операторы равенства и отношения 21
2.4 логические операторы 22
2.5 очередность операторов 23
2.6 ключевые слова 24
2.7 код escape 25
3. СРАВНЕНИЕ
3.1 структура if 26
3.2 структура if .. else 27
3.3 различие между = и == 28
3.4 условный оператор ? 29
3.5 использование файлов 30
4. ЦИКЛ
4.1 цикл for 31
4.2 цикл while 32
4.3 цикл do..while 33
4.4 команда break 34
4.5 команда continue 35
4.6 команда switch 36
4.7 команда exit 37
5. УКАЗАТЕЛЬ
5.1 указатель(pointer) 38
5.2 оператор снятия ссылки “*”(dereference operator) 39
5.3 размер указателя 40
5.4 указатель к указателю 41
5.5 массив 42
5.6 циклы в массиве 43
5.7 многозначный массив 44
5.8 связь между указателем и массивом 45
5.9 динамическая память 46
5.10 вектор 47
5.11 итераторы в векторе 48
5.12 некоторые алгоритмы для вектора 49
6 ПОСЛЕДОВАТЕЛЬНОСТЬ CHARACTER
6.1 последовательность character 50
6.2 размер последовательности character 51

6.3 буфер 52
6.4 манипулятивные функции 53
6.5 класс string 54
6.6 считывание string 55
6.7 конверсия 56
6.8 некоторые string функции 57
7 ФУНКЦИИ
7.1 функции 58
7.2 функции без типа “void” 59
7.3 среда функции 60
7.4 заблаговременная декларация 61
7.5 внешняя функции 62
7.6 ключевое слово namespace 63
7.7 проходя аргументы 64
8 КЛАСС
8.1 класс 65
8.2 уровень доступа 66
8.3 функция доступа 67
8.4 конструктор 68
8.5 перегружая конструктор 69
8.6 стандартный конструктор 70
8.7 деструктор 71
8.8 внешний класс 72
8.9 указатель к классу 73
8.10 указатель “this” 74
9 ПЕРЕГРУЖЕНИЕ И ДРУЖЕСТВЕННОСТЬ
9.1 операторы перегружения 75
9.2 перегружение одинарного оператора 76
9.3 перегружение бинарного оператора 77
9.4 перегружение ввода/вывода оператора 78
9.5 дружественная функция 79
9.6 дружественный класс 80
10 НАСЛЕДОВАНИЕ
10.1 наследование 81
10.2 защищенные члены 82
10.3 связь наследования 83
10.4 инициализация полученных классов 84
10.5 множественное наследование 85
ССЫЛКИ 86

Тема: Простая программа в C++

Это простая C++ программа. Каждая C++ программа содержит функцию int main() { }.При необходимости пользователь включат библиотеку с помощью ее имени и ключевого “#include”. Также в C++ есть Стандартная Библиотека. Для использования ее возможностей следует вводить ключевое слово “std::” для каждой функции. Позже вы узнаете более рациональный способ.

	C++ КОД	РЕЗУЛЬТАТ

1
2
3
4
5
6
7
	#include <iostream>
	
	int main()
	{
	std::cout << "Первая программа";
	return 0;
	}
		Первая программа

Задачи

1. Напишите C++ программу, которая выводит на экран “А вот и вторая программа”
2. Напишите C++ программу, которая выводит на экран “Добро пожаловать в Казахстан”

Тема : Компилирование и запуск

В C++ есть два способа написания программ: с графическим пользовательским интерфейсом (иначе IDE) и с черным экраном (чаще называемым командной панелью , консолью, терминалом). Алгоритм второго способа состоит в следующем:

1 Создайте директорию (папку), затем создайте файл с расширением .cpp и откройте в редакторе (в windows это блокнот).

2 – Напишите C++ код. (также можно скопировать)

3 – Откройте командную панель (в Windows используйте сочетание клавиш Windows sign + R , затем напишите cmd)

4 – дойдите до нужной директории, используя консоль

5 – В консоли напишите “g++ имя_файла.cpp” и нажмите ENTER.

6 – если не возникнет ошибок, появится имя_файла.exe или a.exe

7 – напишите имя файла и нажмите ENTER

	C++ КОД	РЕЗУЛЬТАТ

1
2
3
4
5
6
7
	#include <iostream>
	
	int main()
	{
	std::cout << " Первая программа ";
	return 0;
	}
		Первая программа

Задачи

1. Напишите программу с выводом “Привет”

Тема : Комментарии

В C++ есть 2 типа комментариев:
1 – однострочные комментарии //
2 – многострочные комментарии /* ...
…....
…. */

Большинство начинающих программистов не используют комментарии в своих программах. Однако после большей практики они понимают их важность. Ведь люди имеют склонность забывать.

	C++ КОД	РЕЗУЛЬТАТ
1	//Комментарии в C++	Попытка 1: Введите число: 5 5 нечетное Попытка 2: Введите число: 6 6 четное
2	#include <iostream>	
3		
4	int main()	
5	{	
6	int i; // однострочный комментарий	
7	/* Считывание числа	
8	Для проверки на четность или нечетность	
9	*/	
10	std::cout << "Введите число : ";	
11	std::cin >> i;	
12	//проверка на четность или нечетность	
13	if ((i%2)==0)	
14	std::cout << i <<" четное";	
15	else	
16	std::cout << i <<" нечетное	
17	return 0;	
18	}	

Задачи
1. Напишите C++ программу с комментариями, проверяющая делимость числа на 3
2. Напишите C++ программу , суммирующую два float числа , используя комментарии
3. Напишите C++ программу с комментариями, проверяющая является ли число положительным или отрицательным

Тема : ввод/вывод(в/в)

В C++ есть два наиболее часто используемых операторов ввода/вывода - cin/cout. Для их использования следует включить библиотеку iostream .

В C++ есть Стандартная Библиотека. В ней есть множество готовых функций. Для их использования есть ключевое слово “std::”(чаще называемым namespace) для каждой функции

	C++ КОД	РЕЗУЛЬТАТ

1
2
3
4
5
6
7
8
	#include <iostream> // библиотека для в/в
	int main()
	{
	int i; // обозначен новое integer число
	std::cin >> i; //считывание числа с
	//клавиатуры
	std::cout << i; // вывод числа на экран
	return 0;
		Попытка 1:
		3
		3
		Попытка 2:
		25
		25

Задачи

1. Напишите C++ программу, считывающую два integer числа и показывающая на экране их произведение
2. Напишите C++ программу, считывающую два числа и показывающая их частное
3. Напишите C++ программу, считывающую integer и double и выводящая и сумму

Тема: Типы данных

В C++ есть фундаментальные типы данных для хранения переменных в памяти компьютера. Компьютерная память выражена байтами. Примеры типов данных : Int, long char

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Введите число:5 Введите символ:d 5 d Попытка 2: Введите число:12 Введите символ:Z 12 Z
2		
3	int main()	
4	{	
5	int a;	
6	char c;	
7	std::cout << "Введите число :";	
8	std::cin >> a;	
9	std::cout << "Введите символ :";	
10	std::cin >>c;	
11	std::cout << a << " " << c;	
12	return 0;	
13	}	

Задачи

1. Напишите C++ программу, считывающую два integer числа и выводящая их сумму
2. Напишите C++ программу, считывающую два double числа и выводящая их сумму
3. Напишите C++ программу, считывающую три символа и выводящая их в одну строку

Тема : Тип данных integer

Integer это переменные, используемые для представления целых чисел. В C++ это int, short, long. Также они могут быть обозначенные и необозначенные.

Необозначенные int числа это положительные. По умолчанию integer типы обозначены. Также в C++ int может быть декларирован в одну строку. Например, int a,b,c.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Введите числа: 20 40 Введите long числа: 1000 1 60 1001 Попытка 2: Введите числа: 1 2 Введите long числа: 100000 1 3 100001
2		
3	int main()	
4	{	
5	int a1, a2; long l1,	
6	l2; s1;	
7	std::cout << "Введите числа :";	
8	std::cin >> a1 >> a2;	
9	std::cout << " Введите long числа :";	
10	std::cin >>l1 >> l2;	
11	std::cout << a1 + a2 << " " << l1 + l2;	
12	return 0;	
13	}	
14		

Задачи

1. Напишите C++ программу , выводящую результат частного 2 short чисел.

2. Напишите C++ программу , выводящую результат произведения 3 обозначенных 3 long чисел

3. Напишите C++ программу , выводящую результат произведения long и integer

Тема : Тип данных real numbers

Переменные типа “float”, “double” используются для хранения реальных чисел. Для них “+3.5”, “3.5” и “3.500” имеют одно значение.

В C++ для реальных чисел используется знак “.” , а не “,”.

Реальные числа могут быть написаны с использованием плавающей точки (научная нотация). К примеру, 0.35 может быть написано как “3.5e+01” (т.е. 0.35 x 10).

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2		Введите числа:3.5 6.7
3	int main()	Введите числа: 4.777 -9.76
4	{	10.2 -4.983
5	double a1, a2;	Попытка 2:
6	float l1, l2;	Введите числа:1.1 2.22
7	std::cout << " Введите числа:";	Введите числа: 4 -9
8	std::cin >> a1 >> a2;	3.32 -5
9	std::cout << " Введите числа:";	
10	std::cin >>l1 >> l2;	
11	std::cout << a1 + a2 << " " << l1 + l2;	
12	return 0;	
13	}	

Задачи

1. Напишите C++ программу, выводящую произведение 2 double чисел

2. Напишите C++ программу, выводящую частное 2 float чисел

3. Напишите C++ программу, выводящую сумму 3 float чисел, написанных в научной нотации

Тема : Тип данных bool

Тип данных bool может принимать только два значения: правда и ложь. В типе bool ложь имеет значение 0, а правда равна 1.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2		Введите число:4
3	int main()	Больше 0
4	{	1 0
5	boolb1=false,b2;	Попытка 2:
6	int a1;	Введите число:-4
7	std::cout << " Введите число:";	0 0
8	std::cin >> a1;	
9	if (a1>0) b1=true;	
10	if (b1) std::cout << "Больше 0 \n";	
11	// то же что и (b1==true)	
12	b2=0; //или b2 = false	
13	std::cout << b1 << " " << b2;	
14	return 0;	
15	}	

Задачи

1. Напишите C++ программу, выводящую “101” , используя bool

2. Напишите C++ программу, проверяющую четность или нечетность числа , используя bool

Тема : Тип данных character(символ)

В C++ тип данных char используется для обозначения единичного символа. Есть стандартные символы как заглавные и прописные буквы алфавита, цифры '0'..'9' и некоторые специальные символы, например: *, @, !, и т.д. Собрание символов называется ASCII.

Символы обособляются одинарными кавычками ('). К примеру, в C++ (char c='a'); работает отлично, но (char c="a"); уже является компиляционной ошибкой.

Символы(char) представлены числами(int) . (char это подмножество int). К примеру, в ASCII '9' это обозначается как 57 и 'A' обозначается как 65.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2		Введите символы: 5 7
3	int main()	53 55
4	{	5-7=-2
5	charb1,b2;	Попытка 2:
6	std::cout << "Введите символы :";	Введите символы: Z A
7	std::cin >> b1 >> b2;	90 65
8	std::cout << int(b1) << " " << int(b2) << "\n";	Z-A=25
9	std::cout << b1 << "-" << b2 << "=" <<b1 - b2;	
10	return 0;	
11	}	

Задачи

1. Напишите C++ программу, выводящую символ, соответствующий данному числу.

2. Напишите C++ программу, получающую символ и показывающую его , если это цифра, буква или специальный символ.

Тема : Кастинг типов

Результатом деления двух int чисел является int число. Чтобы в результате получить тип данных double , используется кастинг.

Есть два способа:
ответ = (double) (делимое) / делитель
 ответ = static_cast<double>(делимое) / делитель

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2		Введите числа: 5 2
3	int main()	5/2=2
4	{	5/2=2.5
5	int b1, b2;	5/2=2.5
6	double d1, d2, d3;	Попытка 2:
7	std::cout <<"Введите числа :";	Введите числа: 12 4
8	std::cin >> b1 >> b2;	12/4=3
9	d1 = b1 / b2;	12/4=3
10	d2 = (double) (b1) / b2;	12/4=3
11	d3 = static_cast<double> (b1) / b2;	
12	std::cout << b1 << "/" << b2 << "=" << d1 << "\n";	
13	std::cout << b1 << "/" << b2 << "=" << d2 << "\n";	
14	std::cout << b1 << "/" << b2 << "=" << d3 << "\n";	
15	return 0;	
16	}	

Задачи
1. Напишите C++ программу, получающую два double числа, делящая их на три и затем выведите результат в типе int .

Тема : Среда переменных

Переменная может быть глобальной или местной .

Глобальная переменная это переменная, декларированная в main части исходного кода, вне всех функций, в то время как местная переменная декларирована в пределах главной части или блока. Если имена местной и глобальной совпадают, для глобальной используется ключевое слово “::”.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Введите число:8 местный x1 : 8 глобальный x1 : 5 глобальный x2 : 10 Попытка 2: Введите число:14 местный x1 : 14 глобальный x1 : 5 глобальный x2 : 10
2		
3	intx1=5,x2=10; //глобальные переменные	
4		
5	int main()	
6	{	
7	intx1; //местные переменные	
8	std::cout << "Введите число :";	
9	std::cin >> x1;	
10	std::cout << "местный x1 : " << x1 << "\n";	
11	std::cout << "глобальный x1 : " << ::x1 << "\n";	
12	std::cout << "глобальный x2 : " << x2 << "\n";	
13	return 0;	
14	}	

Задачи

1. Напишите C++ программу, получающая три integer числа. Для каждого добавьте глобальный int x, равный пяти.

Тема : Константы

Переменные константы это переменные с установленным неизменным значением . Их структура: const тип_данных имя_переменной = значение;

Заметка: Переменные константы должны быть инициализированы.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Компиляционная ошибка Попытка 1: строка 5 : ошибка: неинициализированная константа pi
2		
3	int main()	
4	{	
5	const double pi; //ошибка, не	
6	//инициализирована	
7	const int x2 = 5; // нет ошибок	
8	int x3;	
9	std::cout << "Введите радиус :";	
10	std::cin >> x3;	
11	std::cout << "Площадь : " << pi * x3 * x3 << "\n";	
12	return 0;	

Задачи

1. Измените вышеприведенную C++ программу, для ее корректной работы.

2. Напишите C++ программу, вычисляющую объем пирамиды, используя переменные константы.

Тема :Ключевое слово Define

В C++ есть ключевое слово “#define” для обозначения определенных названий часто используемой константы. Иначе это называется макрос. Структура: #define значение identifier

Заметка: чаще всего это используется на олимпиадах для ускорения работы.Не используйте это в больших проектах, т.к. это может вызвать недоступные для понимания ошибки.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Введите радиус: 2 6.18 6.18 Попытка 1: Введите радиус: 1 3.14
2		
3	#define pi 3.14	
4	#define REP(i,n) for(int i=0; i<n; ++i)	
5		
6	int main()	
7	{	
8	int x2 = 5; // нет ошибок	
9	int i,x3;	
10	std::cout << "Введите радиус :";	
11	std::cin >> x3; REP(i,x2)	
12	{	
13	std::cout<<x3*pi << " \n";	
14	}	
15	return 0;	
16	}	
17		

Задачи

1. Напишите C++ программу, вычисляющую площадь круг с обозначенным Пи.

Тема :Ключевое слово Enum

Enumeration (перечисление) для обозначения имени как значения для типа int. К примеру, в неделе 7 дней. Это может быть написано как:

enum Неделя { ПОН=1, ВТН, СРД, ЧТВ, ПТН, СУБ, ВСКР=0}. Этим способом легче сделать понятный код

По умолчанию, членам “перечисления(enum)” задаются значения 0, 1, 2, и т.д., но это может быть инициализировано. Также ошибкой является выражение Week = 1;

	C++ КОД	РЕЗУЛЬТАТ
1	// в субботу в магазине скидка 20%	Попытка1:
2	// в воскресенье 40%	СЕГОДНЯ 0
3	// в другие дни нет скидок	20
4	#include <iostream>	Оплата : 12
5		Попытка 2:
6	int main()	СЕГОДНЯ 0
7	{	24
8	enum Week {MON=1, TUE, WED, THU, FRI, SAT, SUN=0};	Оплата : 14.4
9	Week w;	
10	w = SUN; //предположим, что сегодня	
11	//воскресенье	
12	std::cout << "СЕГОДНЯ " << w << "\n";	
13	int price;	
14	std::cin >> price;	
15	if (w == SAT) std::cout << "Оплата : " << price - price*0.2;	
16	else if (w == SUN) std::cout << " Оплата : " << price - price*0.4;	
17	else std::cout << " Оплата : " << price;	
18	return 0;	
19	}	
20		

Задачи

1. Измените вышеприведенную программу, изменив в ней дни и скидки.

2. Напишите C++ программу для светофора. Используйте перечисление 3 цветов (красный, желтый, зеленый).

Тема : Арифметические операторы

В C++ есть пять стандартных арифметических операторов .

Операция	Арифметический оператор
Сумма	+
Вычитание	-
Произведение	*
Деление	/
Остаток	%

	C++ КОД	КОД
1	// Калькулятор	Попытка 1: Введите два числа: 1 5 Сумма: 6 Попытка 2: Введите два числа : 2 3 Сумма : 5
2		
3	#include <iostream>	
4		
5	int main()	
6	{	
7	int x, y;	
8	std::cout << "Введите два числа : ";	
9	std::cin >> x >> y;	
10	std::cout<<"Сумма:"<<x+y;	
11	return 0;	
12	}	

Задачи

1. Создайте программу- калькулятор для пяти операций

Тема : Краткое написание операторов

Чаще всего для краткого написания используются сложносоставные операторы. Они характеризуются соединением арифметического оператора и assignment оператор (=). К примеру, “x = x + 5;” можно написать как “x += 5;”

Также для краткого написания используются операторы увеличения(++) и уменьшения(++). Например “x = x + 1;” можно заменить на “x++;”.

	C++ КОД	КОД
1	// Калькулятор	Попытка 1: Введите два числа: 4 5 Сумма: 9 Попытка 2: Введите два числа: 3 6 Сумма: 9
2	#include <iostream>	
3		
4	int main()	
5	{	
6	int x, y;	
7	std::cout << "Введите два числа : ";	
8	std::cin >> x >> y;	
9	x+=y;//то же что и x=x+y	
10	std::cout << "Сумма : " << x;	
11	return 0;	
12	}	

Задачи

1. Напишите C++ программу- калькулятор.

Тема : Операторы равенства и отношения

Для решения в C++ есть операторы равенства и отношения:

Операторы отношения	Значение
>	x больше чем y
<	x меньше чем y
>=	x больше или равен y
<=	x меньше или равен y
операторы равенства	
==	x равен y
!=	x не равен y

Заметка: чаще всего они используются с ключевым словом “if”.

	C++ КОД	КОД
1	// Сравнение	Попытка 1: Введите два числа: 1 2 1 меньше чем 2 Попытка 2: Введите два числа: 2 1 2 меньше чем 1
2		
3	#include <iostream>	
4		
5	int main()	
6	{	
7	int x, y;	
8	std::cout << "Введите два числа : ";	
9	std::cin >> x >> y;	
10	if (x > y) std::cout << x << " больше чем" << y;	
11	if (x < y) std::cout << x << " меньше чем " << y;	
12	return 0;	
13	}	

Задачи

1.Измените вышеприведенную программу для операторов отношения.

Тема : Логические операторы

В C++ есть три логических оператора: “и” (&&), “или” (||) и “не” (!).

a	b	a && b
правда	правда	правда
правда	ложь	ложь
ложь	правда	ложь
ложь	ложь	ложь

a	b	a || b
правда	правда	правда
правда	ложь	правда
ложь	правда	правда
ложь	ложь	ложь

Оператор “не” (!) меняет правду на ложь и наоборот

	C++ КОД	КОД
1	// В системе координат есть 4 четверти	Попытка 1:
2	// Используя «И» вычислите, в какой четверти	Введите 2 координаты: 1 1
3	//расположены данные координаты	в 1 четверти
4		Попытка 2:
5	#include <iostream>	Введите 2 координаты: -1 -1
6		в 3 четверти
7	int main()	
8	{	
9	int x, y;	
10	std::cout << "Введите 2 координаты : ";	
11	std::cin >> x >> y;	
12	// в C++ && то же что «И»	
13	if ((x>0) && (y>0)) std::cout << "в 1 четверти";	
14	if ((x<0) && (y>0)) std::cout << " во 2 четверти";	
15	if ((x<0) and (y<0)) std::cout << " в 3 четверти ";	
16	if ((x>0) and (y<0)) std::cout << " в 4 четверти ";	
17	return 0;	

Задачи

1. Напишите C++ программу, получающую 3 integer числа, характеризующие размер треугольника и вычисляющая его тип.

2. Напишите C++ программу, проверяющую является ли данное число n будним или выходным днем.

Тема : Очередность операторов

Как и в алгебре, в C++ есть определенная последовательность операторов (какой оператор будет использован в вычислениях первым). В основном, правила схожи с алгеброй. Простой алгоритм :

Сначала вычисляются операции в скобках. Если выражение состоит из множества скобок, первыми вычисляются внутренние .

Вторыми вычисляются произведение, деление и остаток. Если таких операций несколько, они вычисляются слева направо.

В конце считаются сумма и вычитание. Здесь тоже работает правило «слева направо».

В C++ могут работать даже такие сумасшедшие выражения как“a>>b+a=d%3+(5==3)+c” , при условии существования переменных a,b,c,d.

	C++ КОД	КОД
1	// Очередность операторов	Попытка 1:
2	//в комментариях указана последовательность операций	Введите два числа: 5 7
3	#include <iostream>	192
4		8
5	int main()	Попытка 2:
6	{	Введите два числа: 6 6
7	int x, y;	360
8	std::cout << "Введите два числа : ";	9
9	std::cin >> x >> y;	
10	int z;	
11	z = ((x-y)+(x*x-y)) * (x+y);	
12	// 7 1 4 2 3 6 5	
13	std::cout << z << "\n";	
14	z = ((x==y) + (x%4))+y;	
15	// 5 1 3 2 4	
16	std::cout << z;	
17	return 0;	
18	}	
19		

Задачи

1. Напишите C++ программу, вычисляющую расстояние между двумя точками. Подсказака: используйте функцию sqrt из библиотеки <cmath>

2. Что будет результатом выражения “z = ((x=y)+(x++)) + (x!=y);” и какова последовательность операторов?

Тема : Ключевые слова

В C++ есть определенные ключевые слова . Имя переменной не должно совпадать с именем ключевого слова. C++ чувствителен к регистру, поэтому между return и RETURN есть большая разница.

Список ключевых слов :

asm, auto, bool, break, case, catch, char, class, const, const_cast, continue, default, delete, do, double, dynamic_cast, else, enum, explicit, export, extern, false, float, for, friend, goto,

if, inline, int, long, mutable, namespace, new, operator, private, protected, public, register, reinterpret_cast, return, short, signed, sizeof, static, static_cast, struct, switch, template,

this, throw, true, try, typedef, typeid, typename, union, unsigned, using, virtual, void, volatile, wchar_t, while

Добавочно :

and, and_eq, bitand, bitor, compl, not, not_eq, or, or_eq, xor, xor_eq

	C++ КОД	КОД

1
2
3
4
5
6
7
8
9
10
11
12
	#include <iostream>
	
	int main()
	{
	int x, y;
	std::cout << "Введите два числа : ";
	std::cin >> x >> y;
	int for;
	for = x + y;
	std::cout << for;
	return 0;
	}
		Попытка 1:
		Компиляционная ошибка

Задачи
1. Измените вышеприведенную программу для ее корректной работы

Тема : Код Escape

В C++ есть escape коды, используемые для выражения большего количества символов и действий. Структура кода escape: знак “\” и буква.

\n	новая линия
\r	carriage return
\t	tab
\v	вертикальный tab
\b	backspace
\f	form feed (page feed)
\a	звуковой сигнал (beep)
\'	Одинарные кавычки (')
\"	Двойные кавычки (")
\?	Вопросительный знак (?)
\\	backslash (\)

	C++ КОД	КОД
1	#include <iostream>	Попытка 1:
2		Введите 2 символа :
3	int main()	a W
4	{	a ? ' W
5	char ch1, ch2;	Попытка
6	std::cout << "Введите 2 символа : ";	2:
7	std::cin >> ch1 >> ch2;	Введите 2 символа:
8	std::cout << "---------- \n";	2 z
9	std::cout << ch1 << "\t \? \' \t" << ch2;	2 ? ' z
10	return 0;	
11	}	

Задачи

1. Напишите C++ программу, выводящая текст в двойных кавычках.

Тема : Ключевое слово if

В большинстве языков программирования есть ключевое слово решения «if». Оно используется для проверки условия. Структура:

if (условие или условия) C++ код ИЛИ

if (условие или условия)

{ C++ код }

Для более усложненных условий могут использоваться логические операторы

	C++ КОД	КОД
1	#include <iostream>	Попытка 1: Введите символ: a Вы ввели символ a Попытка 2: Введите символ: d Вы не ввели символ a
2		
3	int main()	
4	{	
5	char ch1;	
6	std::cout << "Введите символ : ";	
7	std::cin >> ch1;	
8	if (ch1=='a') std::cout<< "Вы ввели символ a";	
9	if (ch1!='a') std::cout<< "Вы не ввели символ a";	
10	return 0;	
11	}	

Задачи

1. Напишите C++ программу, проверяющую четность или нечетность числа

2. Напишите C++ программу ,проверяющую является ли число двухзначным или трехзначным. (Подсказка: используйте оператор остатка “%”)

Тема : структура if … else

В C++ для решения может быть использована структура if .. else. Структура :

if (условие) код

else if (условие) код

else код

Заметка: если в коде два и более утверждения , используйте скобки “{}”

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Введите символ: a Вы ввели букву И она прописная Попытка 2: Введите символ: ‘ Вы ввели символ
2		
3	int main()	
4	{	
5	char ch1;	
6	std::cout << "Введите символ : ";	
7	std::cin >> ch1;	
8	if (ch1>='a' and ch1<='z')	
9	{	
10	std::cout<< "Вы ввели букву \n";	
11	std::cout<< "И она прописная";	
12	}	
13	else if (ch1>='A' && ch1<='Z')	
14	std::cout<< "Вы ввели заглавную	
15	букву";	
16	else if (ch1>='0' && ch1<='9') std::cout<<	
17	"Вы ввели цифру";	
18	else	
19	{	
20	std::cout<< Вы ввели символ";	
21	char ch2 = ch1;	
22	}	
23	return 0;	

Задачи

1.Напишите C++ программу, проверяющую лежит ли данная точка в треугольнике или вне его

2. Напишите C++ программу, проверяющую равны ли данные квадраты. Если нет, который из них больше.

Тема : Разница между = и ==.

Чаще всего программисты, начинающие изучение C++ имеют трудности с пониманием операторов = и ==.

Этот тип логических ошибок чаще встречается в структуре if. Предположим, что в C++ коде написано: “if (n=5) ” и“if (n==5)”.

В первом случае n приравнивается к числу. (Теперь число n равно 5)

Во втором случае число n сравнивается с 5 и возвращает 1 (правда)

или 0 (ложь).

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Введите два числа: 2 3 2 не равно 5 5 равно 5 Попытка 2: Введите два числа: 9 9 9 не равно 5 5 равно 5
2		
3	int main()	
4	{	
5	int n1, n2;	
6	std::cout << "Введите два числа: ";	
7	std::cin >> n1 >> n2;	
8	if (n1==5) // проверяет n1 и 5	
9	std::cout<< n1 << " равно 5 \n"; else	
10	std::cout<< n1 << " не равно 5 \n";	
11	if (n2=5) // приравнивает n2 к 5	
12	std::cout<< n2 << " равно 5 \n"; else	
13	std::cout<< n2 << " не равно 5 \n";	
14	return 0;	
15	}	
16		
17		

Задачи

1. Напишите C++ программу, проверяющую является ли год високосным для данного n

2. Напишите C++ программу, проверяющую является ли данный символ заглавной 'A' или нет.

Тема : Условный оператор ?

В C++ можно использовать условный оператор “?” вместо структуры if .. else. Структура :

условие ? результат1 : результат2;

Также условный оператор может быть использован в потоке cout, очень мощной возможности C++.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Введите два числа: 1 4 4 is максимум 1 is минимум Попытка 2: Введите два числа: 8 2 8 is максимум 2 is минимум
2		
3	int main()	
4	{	
5	int n1, n2;	
6	std::cout << "Введите два числа : ";	
7	std::cin >> n1 >> n2;	
8	int max;	
9	max=(n1>n2)?n1:n2;	
10	std::cout<< max <<" максимум\n";	
11	std::cout<<((n1<n2)?n1:n2)<<минимум";	
12	return 0;	
13	}	

Задачи

1. Напишите C++ программу, вычисляющую максимум для 3 чисел, используя условного оператора

Тема : Использование файлов

Для ввода и вывода рациональнее всего использовать файлы . Писать одни и те же переменные снова и снова для проверки программы является лишней тратой времени. Программист может совмещать. К примеру, считывать с файла, а результат выводить на экран.

Структура:

#include <fstream>
using namespace std;
 	
int main()
{
ifstream fin("имя_файла");
ofstream fout("имя_файла ");
fin>>некоторые_данные;
fout<< некоторые_данные;
fin.close();
fout.close();
return 0;
}
	ПРАВИЛА
	1 Вместо <iostream>используется <fstream>
	2 В папке с программой должен быть создан файл ввода
	3. Вместо cin, cout используется fin, fout
	4. В конце закрывайте файлы.
	
	ЗАМЕЧАНИЕ: Не забывайте, что программа считывает все с файла и теперь нет необходимости писать что-то вроде «Введите два числа»

Замечание: Вместо использования std::fin или std::cin можно написать using namespace std. Это более рационально.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <fstream>	Попытка 1: в ввод.txt файле 3 4 в вывод.txt файле 4 максимум 3 минимум
2	using namespace std;	
3		
4	int main()	
5	{	
6	int n1, n2;	
7	ifstream fin("ввод.txt");	
8	ofstream fout("вывод.txt");	
9	fin >> n1 >> n2;	
10	int max;	
11	max = (n1 > n2) ? n1 : n2;	
12	fout<< max <<" максимум \n";	
13	fout<< ((n1<n2) ? n1:n2) << " минимум";	
14	fin.close();	
15	fout.close();	
16	return 0;	
17	}	

Задачи

1. Напишите C++ программу, считывающую с файла “цифра.txt” 15 чисел и сохраняющая их сумму в “вывод.txt”

Тема :Цикл for

Структура цикла“for”:

for (инициализация; условие; увеличение)

{ C++ код }

Это работает в четыре шага. Первый и второй шаги заканчиваются “;”. Шаги:

1 – запускается инициализация. Запускается лишь раз.

2 – проверяется условие. Если оно правдиво, цикл продолжается, иначе цикл останавливается и код не запускается.

3 – запускается код. Если количество утверждений больше двух, они должны быть заключены в “{}” скобки.

4 – запускается увеличение и цикл перемещается на второй шаг.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	5
3		1 2
4	int main()	2 4
5	{	3 6
6	int n1, n2;	4 8
7	cin >> n1;	5 10
8	for (int i=1; i<=n1; i++)	Попытка 2:
9	{	3
10	cout << i << " " << i*2;	1 2
11	cout << endl; // новая линия, то же что и cout<<"\n";	2 4
12	}	3 6
13	return 0;	
14	}	

Задачи

1. Напишите C++ программу, вычисляющую сумму первых n элементов

2. Напишите C++ программу, показывающую числа от n до 1.

3. Напишите C++ программу, выводящую все символы. Подсказка: символы- это числа в ASCII коде.

Тема : Цикл while

Структура цикла while : while (условие(я) правда) { код }

В цикле while должны быть операции увеличения (уменьшения) или другие операции, делающие условия ложными. Иначе это будет бесконечный цикл.

Разница между for и while состоит в следующем : для for цикла количество шагов конечно, а в while число шагов зависит от условия.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: 135 531не палиндром 135 Попытка 2: 135531 135531 палиндром 135531 Попытка 1: 1234567654321 -1126087180 не палиндром 2147483647 (ЗАМЕТКА: возникает ошибка, т.к. иапазон данного числа больше диапазона long int)
2	using namespace std;	
3		
4	int main()	
5	{	
6	long int n1, n2, n3=0;	
7	cin >> n1;	
8	n2 = n1;	
9	while (n1!=0)	
10	{	
11	n3 = n3 * 10 + (n1%10);	
12	n1 = n1 / 10;	
13	}	
14	if (n3 == n2)	
15	cout << n3 << палиндром " << n2;	
16	else	
17	cout << n3 << "не палиндром " << n2;	
18	return 0;	
19	}	

Задачи

1. Напишите C++ программу, вычисляющую количество цифр в данном числе n.

2. Напишите C++ программу, проверяющую является ли число prime числом.

3. Напишите C++ программу, выводящую все делители данного числа

Тема :Цикл do … while

Структура цикла “do .. while”: do { код } while (условие[я] правда);

Это работает как цикл“while”. Если программист забудет поменять условия на «ложь», может произойти бесконечный цикл.

Главная разница между циклами do/while и while loops состоит в том, что иногда в do/while может быть еще один шаг, т.к сначала запускается код, потом проверяется условие.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	15
3		AAAAA
4	int main()	AAAAA
5	{	
6	long int n1, n2, n3=0;	Попытка 2:
7	cin >> n1;	5
8	n2 = 10;	A
9	do	
10	{	
11	cout << "A";	
12	n2++;	
13	} while (n2<n1);	
14		
15	cout << endl;	
16	n2 = 10;	
17	while (n2<n1)	
18	{	
19	cout << "A";	
20	n2++;	
21	}	
22	return 0;	
23	}	

Задачи

1. Напишите C++ программу, выводящую все prime числа до заданного числа

2. Напишите C++ программу, выводящую числа до указанного

числа n=3 именно так, как указано ниже.

1

22

333

Тема: Команда break

Команда Break используется для завершения цикла и выхода из него, несмотря даже на правдивость условий.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	8
3		8 7 6 5 4 Пауза
4	int main()	Попытка 2:
5	{	11
6	int n1;	11 10 9 8 7 6 5 4 Пауза
7	cin >> n1;	
8	for (int i=n1; i>0; i--)	
9	{	
10	cout << i << " ";	
11	if (i==4)	
12	{	
13	cout << "Пауза";	
14	break;	
15	cout << "Стоп";	
16	}	
17	}	
18	return 0;	
19	}	

Задачи

1. Напишите C++ программу, находящую наибольшее prime число для данного числа, используя break.

2. Напишите C++ программу, получающую два числа (x, y) и выводящая все числа между ними. Цикл останавливается при условии существования числа от x до y , равному их сумме.

Тема : Команда continue

Команда continue заставляет пройти оставшийся цикл и идти к следующей итерации.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	9
3		9 8 7 6 5 Продолжить 3 2 1
4	int main()	Попытка 2:
5	{	11
6	int n1;	11 10 9 8 7 6 5 Продолжить 3 2 1
7	cin >> n1;	
8	for (int i=n1; i>0; i--)	
9	{	
10	if (i==4)	
11	{	
12	cout << "Продолжить ";	
13	continue;	
14	cout << "Стоп ";	
15	}	
16	cout << i << " ";	
17	}	
18	return 0;	
19	}	

Задачи

1. Напишите C++ программу, выводящую все нечетные числа до данного числа

Тема : Команда switch

Команда “switch” это структура if .. else в более легком варианте. Структура :

```
switch ( переменная ) {
  case 1 :
    // Процесс для переменной = 1
    ...
    break;

  case 5 :
    // Процесс для переменной = 5
    ...
    break;

  default :
    // Процесс для остальных случаев
    ...
}
```

Заметка: после каждого случая нужно использовать команду break, в противном случае идет перход к другим случаям(см. Попытку 3 в примере)

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	Введите свою оценку: 3
3		Плохо
4	int main()	Попытка 2:
5	{	Введите свою оценку: 4
6	int n1;	Хорошо
7	cout << "Введите свою оценку : ";	Попытка 3:
8	cin >> n1;	Введите свою оценку: 5
9	switch (n1)	Очень хорошо
10	{	
11	case 5 : cout << "Очень хорошо";	
12	//команда break;	
13	case 4 : cout << " Хорошо "; break;	
14	case 3 : cout << "Плохо"; break;	
15	case 2 : cout << "Очень плохо"; break;	
16	default: cout << "Вводите число от 2 до 5"; break;	
17	}	
18	return 0;	
19	}	

Задачи

1. Напишите C++ программу, выводящую имя месяца в соответствии с данным числом.

Тема : Команда exit

Команда “exit” используется для немедленного прекращения работы программы. Ее можно использовать когда в коде случается ошибка.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <cstdlib>	Введите два числа :
3	using namespace std;	6 3
4		2
5	int main()	Попытка 2:
6	{	Введите два числа:
7	int n1, n2;	6 0
8	cout << "Введите два числа : ";	
9	cin >> n1 >> n2;	
10	if(n2==0)exit(1);	
11	int div = n1 / n2;	
12	cout << div;	
13	return 0;	
14	}	

Задачи

1. Напишите C++ программу, вычисляющую квадратный корень , используя delta. Если delta меньше 0, программа должна немедленно остановиться.

Тема : Указатель(pointer)

Указатель это переменная, содержащая адрес другой переменной. Указатель можно сравнить с пультом от телевизора . Структура:

тип_данных * имя_указателя;

Знак адреса оператора (&) используется для получения адреса переменной.

	C++ КОД	РЕЗУЛЬТАТ

1
2
3
4
5
6
7
8
9
10
11
12
13
14
	#include <iostream>
	using namespace std;
	int main()
	{
	int intStandart;
	int*intPointer;
	cin >> intStandart;
	intPointer=&intStandart;
	cout <<"Значение"<< intStandart << endl;
	cout <<"Адрес стандарта" << &intStandart << endl;
	cout<<"Значение указателя"<< intPointer << endl;
	cout << "Адрес указателя" << &intPointer << endl;
	return 0;
	}
		Попытка 1:
		12
		Адрес стандарта 0xbffc666c
		Значение указателя 0xbffc666c
		Адрес указателя 0xbffc6668
		
		Попытка 2:
		22
		Адрес стандарта 0xbffc666c
		Значение указателя 0xbffc666c
		Адрес указателя 0xbffc6668
		
		Заметка: адрес может быть изменен в зависимости от системы

Задачи:

1. Напишите C++ программу, совершенствующую вышеприведенную программу использованием double чисел.

2. Напишите C++ программу, получающую число integer x и создающую 2 указателя ptr1 и ptr2 для integer x. Выведите значения x , ptr1 и ptr2.

Тема : оператор снятия ссылки “*”(Dereference operator)

Указатель это адрес другой переменной. Чтобы показать адрес указанной (pointed) переменной, используется оператор снятия ссылки “*”.

Не путайте знак * с оператором снятия ссылки и указателем. Оператор снятия ссылки используется после декларирования указателя.

	C++ КОД	РЕЗУЛЬТАТ

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
	#include <iostream>
	using namespace std;
	int main()
	{
	int a;
	int *ptr1=&a, *ptr2=&a; // здесь * использован для указателя
	
	pointer cin >> a;
	
	cout << "Значение " << a << endl;
	cout << " Значение " << *ptr1 << endl;
	
	*ptr2 = 2 * a; // здесь* использован как снятия ссылки
	cout << " Значение после изменения" << *ptr1 << endl;
	
	return 0;
	}
		Попытка 1:
		24
		Значение 24
		Значение 24
		Значение после изменения 48
		
		Попытка 2:
		4
		Значение 4
		Значение 4
		Значение после изменения 8

Задачи

1. Напишите C++ программу, меняющую два integers числа, используя указатель.

2. Напишите C++ программу, показывающую сумму 2 чисел ,используя указатель.

3 Напишите C++ программу, получающую integer x. Затем сделайте указатель к x и поменяйте его значение. Покажите значение x, значение указателя , значение указателя, используя снятие ссылки.

ТЕМА: Размер указателя

Размер указателя 4 или 8 байтов, в зависимости от архитектуры компьютера. Размер не меняется от типа данных указателя. Напоминание: “указатель сохраняет адрес переменной, а не ее значение ”.

	C++ КОД	РЕЗУЛЬТАТ

1
2
3
4
5
6
7
8
9
10
11
12
13
14
	#include <iostream>
	using namespace std;
	
	int main()
	{
	char a;
	double b;
	char *ptr1=&a; // Указатель для integer
	double *ptr2=&b; // Указатель для //double
	
	cin >> a;
	cin >> b;
	
	cout << sizeof(a) << endl; // размер char
	cout << sizeof(ptr1) << endl; // размер
	//указателя
	cout << sizeof(b) << endl; // размер double
	cout << sizeof(ptr2) << endl; // размер
	//указателя
		Попытка 1:
		a
		1.3
		1
		4
		8
		4
		
		Попытка2:
		Q
		2
		1
		4
		8
		4

Задачи

1. Напишите C++ программу, выводящую размеры типов данных int, long int, float и размеры их указателей

ТЕМА: Указатель к указателю

В C++ есть понятие указателя к указателю. Это означает, что если существует указатель, программист может сделать другой указатель к нему. В этом случае добавьте звездочку (*) для каждого уровня ссылки в декларации. Структура:

тип_данных *указатель1;

тип данных **указатель2 = &указатель1; и т.д.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	24
3		Значение 1 : 24
4	int main()	Адрес1: 0xbfb4e9ec
5	{	Значение2: 0xbfb4e9ec
6	int a; //integer число a	Адрес2: 0xbfb4e9e8
7	int *ptr1=&a; // указатель к a	Значение3 : 0xbfb4e9e8
8	int**ptr2=&ptr1;//указатель к указателю1	Адрес3: 0xbfb4e9e4
9		
10	cin >> a;	Замечание: в тестировании
11		адрес может быть разным
12	cout << "Значение1 : " << a << " Адрес1 : " << &a << endl;	
13	cout << " Значение2 : " << ptr1 << " Адрес2: " << &ptr1 << endl;	
14	cout << " Значение3 : " << ptr2 << " Адрес3 : " << &ptr2 << endl;	
15		
16	return 0;	
17	}	

Задачи

1. Напишите C++ программу, получающую типы данных char, double и long int . Затем сделайте тройной указатель к указателю.

Тема : Массив

Массив это тип данных, сохраняющий многочисленные однотипные переменные через одинаковое имя с использованием индекса. Индекс это параметр integer числа с использованием подсценарного (subsript) оператора([]). D C++ индекс начинается с 0 и заканчивается размером массива - 1 элемент. Индекс используется для присвоения i^{oro} -1 элемента массива.

Струтура :

Тип_данных имя_массива [размер_массива];

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	1
3		2
4	int main()	3
5	{	4
6	int a1,a2,a3,a4,a5; //5 integer чисел	5
7	inta[5]; //массив для 5 чисел int	5 4 3 2 1
8		
9	for (int i=0; i<5; i++)	Попытка 2:
10	cin >> a[i];	5
11		4
12	for (int i=4; i>=0; i--)	3
13	cout << a[i] << " ";	2
14		1
15	return 0;	1 2 3 4 5
16	}	

Задачи

1. Напишите C++ программу, создающую массив из 20 integer чисел, считывающую 20 integer чисел и показывающую нечетные, четные, положительные, отрицательные числа и затем единственное число в нечетном индексе .

2. Напишите C++ программу, создающую массив из 20 double чисел, считывающую 20 double чисел и находящую максимум, минимум и затем выводящую их индексы.

Тема : Циклы в массиве

Большинство массивов используются с циклами (for цикл, while цикл). С ними используются такие операции как:

1) Поиск значения (поиск x в массиве).
2) Вычисление значения (среднее значение, итого массива)
3) Реорганизация массива (сортировка от наименьшего к наибольшему)

В массиве могут быть ошибки как «вне границы»,означающие что программист пытается вызвать индекс больше чем он может быть. К примеру, есть массив из 5 элементов. Программист пишет массив [5]и это ошибка, потому что 5ый элемент массива это массив[4].

Для предотвращения таких ошибок есть лишь один способ. Это практика. .

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	3
3		4
4	int main()	2
5	{	1
6	int size=5;	5
7	int a[size]; // массив для 5 integer чисел	1 2 3 4 5
8		
9	for (int i=0; i<size; i++)	Попытка 2:
10	cin >> a[i];	10
11		9
12	for (int i=0; i<size-1; i++)	8
13	{	7
14	for (int j=i+1; j<size; j++)	6
15	if (a[i]>a[j])	6 7 8 9 10
16	swap(a[i], a[j]);	
17	// swap функция для обмена значениями между двумя	
18	данными	
19	}	
20	for (int i=0; i<size; i++)	
21	cout << a[i] << " ";	
22		
23	return 0;	
	}	

Задачи

1. Напишите C++ программу, сортирующую 20 чисел от наибольшего к наименьшему(убывающий порядок)

2. Напишите C++ программу, имеющую 3 массива double чисел a,b,c. Массивы a и b должны быть считаны с файла. Их сумма сохраняется в массиве c. Выведите c на экран

Тема : Многозначный массив

Многозначный массив это массив, содержащий "массив массивов". К примеру, массив матрицы может быть представлен как таблица элементов с данным размером, как указано ниже.
Структура:

Тип_данных имя_массива [число_рядов][число_столбцов] // int a[2][3]

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	2 2
3	#define width 5	1
4	#define height 3	3
5	int main()	4
6	{	5
7	int a,b;	1 3
8	cin >> a >> b;	4 5
9	intmatrix[a][b]; //многозначный массив	1 1
10		-1 5
11	for (int i=0; i<a; i++)	
12	for (int j=0; j<b; j++)	Попытка 2:
13	cin >> matrix[i][j];	3 2
14		9
15	for (int i=0; i<a; i++) {	9
16	for (int j=0; j<b; j++)	9
17	cout << matrix[i][j] << " ";	9
18	cout<<endl; // зачем это здесь?	9
19	}	9
20		9 9
21	for (int i=0; i<a; i++)	9 9
22	for (int j=0; j<b; j++)	9 9
23	if (i>j) matrix[i][j]=-1;	9 1
24	else if (i<j) matrix[i][j]=1;	-1 9
25		-1 -1
26	for (int i=0; i<a; i++) {	
27	for (int j=0; j<b; j++)	
28	cout << matrix[i][j] << " ";	
29	cout<<endl; // зачем это здесь? }	
30	return 0;	
	}	

Задачи

1. Напишите C++ программу, умножающую две матрицы (2x2)

2. Напишите C++ программу, выводящую таблицу умножения

Тема: Связь между указателем и массивом

В C++ указатель может быть равен адресу первого элемента в массиве. Это можно представить как :

```
1  int a [20];
2  int *ptr1;
3  ptr1 = a; // ptr1 = &a[0]
```

Программист может использовать арифметику чтобы перейти на следующий элемент(ptr1++). Замечание: C+ компилятор автоматически переходит к следующему элементу в зависимости от типа данных. Это означает что если массив состоит из double чисел, тогда размер указателя 8 байтов.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	a X !
3		2
4	int main()	4
5	{	8
6	char ch[3]={'a','X','!'};	0 2
7	char *ptr2;	1 4
8	ptr2 = ch;	2 8
9	cout << *ptr2 << " ";	
10	cout << *(++ptr2) << " ";	Попытка 2:
11	cout << *(++ptr2) << " " << endl;	a X !
12		4
13	int a[3];	16
14	int *b;	34
15	int x;	0 4
16	b = a; // or b = &a[0];	1 16
17	cin >> x ;	2 34
18	*b = x;	
19	cin >> x;	
20	b++;	
21	*b = x;	
22	cin >> x;	
23	b = a;	
24	*(b+2) = x;	
25	for (int i=0; i<3; i++)	
26	cout << i << " " << a[i] << endl;	
27		
28	return 0;	
29	}	

Задачи

1. Напишите C++ программу, считающую четные integer числа в массиве, используя указатель. Используйте : for (int *ptr = a; ptr < a +3; ptr++) ,где «a» это массив и 3 это размер массива

Тема : Динамическая память

В большинстве программ размеры памяти обозначены до запуска программы. Если программисту необходимо изменить размер памяти в период запуска, используется динамическая память с операторами new и delete. Выделение динамической памяти позволяет выделять память любого размера во время работы программы.

int *ptr = new int; // динамически выделяет integer
*ptr = 7; // присваивает 7 к этому integer
delete ptr; // удаляет присвоенную память, приравненной к ptr значению
Замечание: оператор delete не удаляет указатель , он удаляет память, на которую указывает указатель.

Для массивов используются new[], delete[]
int *ptr = new int[5]; // динамически выделяет массив с гибким размером
*ptr[2] = 7; // присваивает 7 ко второму
элементу массива
delete[] ptr; // не путайте с delete

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	4
3		1
4	int main()	2
5	{	3
6	int*a=newint;	4
7	cin >> *a;	1 2 3 4
8		
9	int*b=newint[*a];	Попытка 2:
10	for (int i = 0; i<*a; i++)	3
11	cin >> b[i];	-1
12		-6
13	for (int i = 0; i<*a; i++)	8
14	cout << b[i] << " ";	-1 -6 8
15		
16	delete a;	
17	delete[] b;	
18	return 0;	
19	}	

Задачи
1. Напишите C++ программу, получающую integer n и создающую динамический массив double чисел размера n. Затем выведите максимум в массиве. Потом удаляется массив a ,опять вводится integer n и создается динамический массив a размера n long int чисел. В конце покажите минимум в массиве.

Тема : Вектор

Векторы представлены как динамические массивы. Элементы вектора хранятся в непрерывном хранении и имеют уровень доступа как у массива. Векторы имеют одно важное преимущество в сравнении с массивами в C++: у вектора можно изменить размер во время работы программы для размещения дополнительных элементов.

Структура:

vector<тип_данных> имя_вектора; // или vector<тип_данных> имя_вектора (размер);

Замечание:: для использования вектора используйте в коде #include <vector>

Для изменения размера вектора программист может использовать функцию resize(size), push_back(новый_элемент).

Также в стандартной библиотеке для вектора есть готовые функции для облегчения программирования. Некоторые из них представлены ниже:

begin(), end()	Начало или конец вектора
size()	Считает количество элементов в векторе
resize(int)	Увеличивает или уменьшает размер вектора
push_back, pop_back	Добавляет или удаляет элемент из вектора

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <vector>	3
3	using namespace std;	1
4		3
5	int main()	5
6	{	Размер 1 : 3
7	vector<int>v1;	Размер 2 : 5
8	int a;	1 3 5 80
9	cin >> a;	
10	v1.resize(a);	Попытка 2:
11		4
12	for (int i=0; i<v1.size(); i++)	-1
13	cin >> v1[i];	7
14	cout << "Размер 1 : " << v1.size() << endl;	-4
15	v1.push_back(80);	5
16	v1.push_back(90);	Размер 1 : 4
17	cout << " Размер 2 : " << v1.size() << endl;	Размер 2 : 6
18	v1.pop_back();	-1 7 -4 5 80
19	for (int i=0; i<v1.size(); i++)	
20	cout << v1[i] << " ";	
21		
22	return 0;	
23	}	

Задачи

1. Напишите C++ программу, создающую GPA_студентов и получающая GPA по номеру студентов в классе. Затем получите среднее арифметическое класса.

2. Напишите C++ программу, получающую вектор n double чисел и сортирующая их.

Тема : Итераторы в векторе

Итераторы обеспечивают гибкий способ доступа к элементам в векторе. Они позволяют ссылаться сразу на элементы в векторе и итерировать через весь вектор.
Структура : vector<int>::iterator i;
Как и в теме «указатель» , для доступа к указанному итератором элементу используется оператор «*»

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <vector>	3
3	using namespace std;	1
4		2
5	int main()	3
6	{	1 2 3 80
7	vector <int> v1;	
8	vector<int>::iterator i;	Попытка 2:
9	int a;	2
10	cin >> a;	-4
11	v1.resize(a);	5
12		-4 5 80
13	for (i=v1.begin(); i<v1.end(); i++)	
14	cin >> *i;	
15		
16	v1.push_back(80);	
17		
18	for (i=v1.begin(); i<v1.end(); i++)	
19	cout << *i << " ";	
20		
21	return 0;	
22	}	

Задачи
1. Напишите C++ программу, использующую итераторы для нахождения максимального элемента в векторе

2. Напишите C++ программу, удаляющую всю информацию в векторе

Тема : Некоторые алгоритмы для вектора

В большинстве языков программирования алгоритмы реализованы в Библиотеке. Иначе, «не изобретайте велосипед». В Стандартной Библиотеке C++ есть алгоритмы и для вектора.

Алгоритм	Описание и пример использования
find	Находит данное значение. Если значение найдено, Алгоритм возвращает итератор , иначе возвращает end. if (find (v1.begin(), v1.end(), 25) == v1.end())
sort	Сортирует элементы в векторе sort (v1.begin(), v1.end());

reverse	Меняет порядок элементов в векторе
	sort (v1.begin(), v1.end());
	reverse (v1.begin(), v1.end());

replace	Заменяет все данные случаи другими replace(v1.begin(), v1.end(), 'a', 'A');

Замечание: To use these algorithms in C++ код #include <algorithm>
Замечание: Для более подробной информации просто прогуглите это.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <vector>	3
3	#include <algorithm>	1
4	using namespace std;	2
5		3
6	int main()	1 2 3
7	{	3 2 1
8	vector<int>v1;	
9	int a;	Попытка 2:
10	cin >> a;	4
11	v1.resize(a);	-4
12		5
13	for (int i= 0; i<v1.size(); i++)	-3
14	cin >> v1[i];	6
15		-4 -3 5 6
16	sort(v1.begin(),v1.end());	6 5 -3 -4
17	for (int i= 0; i<v1.size(); i++)	
18	cout << v1[i] << " ";	
19		
20	cout << endl;	
21	reverse(v1.begin(),v1.end());	
22	for (int i= 0; i<v1.size(); i++)	
23	cout << v1[i] << " ";	
24		
25	return 0;	
26	}	

Задачи
1. Напишите C++ программу, которая ищет определенное число в векторе

Тема : Последовательность Character

Последовательность Character (массив символов) создает string. String это последовательность символов(character), которые переводятся как часть текста. Для обозначения конца последовательности используется специальный символ: *null character* '\0' (backslash, ноль).
char str[]="SDU";
char str [] = { 'S', 'D', 'U', '\0' };

0	1	2	3
S	D	U	\0

Этот тип называют Ctyle(Стиль си) string. В C++ есть специальный класс <string>.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	Введите имя:Аскар
3		Пока, Аскар!
4	int main ()	
5	{	Попытка 2:
6	char init[] = "Введите имя:";	Введите имя:А
7	char out[] = "Пока, ";	Пока, А!
8	char name[25];	
9	cout << init;	
10	cin >> name;	
11	cout << out << name << "!";	
12	return 0;	
13	}	

Задачи
1. Напишите C++ программу, считывающую имя и фамилию и затем выводящую их раздельно

2. Напишите C++ программу, считывающую имя, возраст и класс студента, затем выводящая их

Тема : Размер последовательности Character

Размер символов равен :текст плюс 1 (null operator).
char str[]="SDU";

0	1	2	3
S	D	U	\0

Но если мы декларируем как [integer] с данным размером, тогда размер показывается полным

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	Введите имя: Аскар
3		13
4	int main ()	E n t e r n a m e :
5	{	25
6	char init[] = "Введите имя :";	A c к a p $ e ^ & # @
7	char name[25];	
8	cout << init;	Попытка 2:
9	cin >> name;	Enter name : Аскар Алматы
10		13
11	cout << sizeof(init)<<endl;	E n t e r n a m e :
12	for (int i=0; i < sizeof(init); i++)	25
13	cout << init[i] << " ";	A c к a p $ e ^ & # @
14		
15	cout << endl <<sizeof(name)<<endl;	Замечание: символы могут быть
16	for (int i=0; i < sizeof(name); i++)	другими
17	cout << name[i] << " ";	
18		
19	return 0;	
20	}	

Задачи
1. Напишите C++ программу, считывающую имя и выводящую только гласные

Тема : Буфер

Буфер – это временная память для хранения данных. Так в C++ с помощью «cin» Программист временно держит вводимую информацию пользователя. Если пользователь введет информации больше чем может вместить массив, случится ошибка. Это называют переполнением буфера. Переполнение буфера записывается в другой памяти, будучи переписанным. Это в свою очередь вызывает поломку программы или любое количество других случаев

Рекомендуемый способ считывания типа string это использование cin:
cin. getline (имя _ массива, число_ символов);

Вызов cin.getline() будет считывать до: число_символов - 1 символы в имя _ массива (1 это null завершитель!). Любые лишние символы будут удалены. Это гарантирует отсутствие проблемы переполнения буфера.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	Введите имя: Аскар
3		А С К А Р $ e ^ & # @
4	int main ()	
5	{	Попытка 2:
6	char init[] = "Введите имя :";	Введите имя: Аскар Алматы
7	char name[25];	А С К А Р А Л М А Т Ы ^ & #
8	cout << init;	@
9	cin.getline(name,25);	
10		Замечание: символы могут меняться
11	for (int i=0; i < sizeof(name); i++)	
12	{	
13	if ((name[i]>=97) && (name[i]<=122))	
14	cout << char(name[i]-'a'+'A') << " ";	
15	else cout << name[i] << " ";	
16	}	
17		
18	return 0;	
19	}	

Задачи
1. Напишите программу, считывающую имя и фамилию вместе с пробелом и выводящую только имя

2. . Напишите программу, считывающую некоторый текст и конвертирующую все заглавные буквы в прописные и наоборот

3. Напишите программу, считывающую 3 города с их населением в одной строке и затем выводящую город с максимальным населением.

Тема : Манипулятивные функции

В C++ есть много функций для манипуляций с Cstyle strings.

strcat(destination, source)	Соединяет один string с другим
strncat(destination, source, n)	Соединяет один string с другим с длиной n
strcmp(destination, source)	Сравнивает два string (если равны, возвращает 0)
strncmp(destination, source, n)	Сравнивает два string с определенным числом символов (если равны, возвращает 0)
strlen(string)	Возвращает длину string (исключая null terminator)
strcpy(destination, source)	Копирует один string в другой.
strncpy(destination, source, n)	Копирует один string в другой с длиной n.

Замечание: для их использования включайте в код #include <cstring>

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <cstring>	Введите имя: Аскар
3	using namespace std;	Аскар
4		Аск
5	int main ()	А с к а р
6	{	
7	char init[] = "Введите имя :";	Попытка 2:
8	char name[25]; cout << init;	Введите имя:
9	cin.getline(name, 25);	Аскар А
10		Аскар А
11	char last[40];	Аск
12	strcpy(last,name);	А с к а р А
13	cout << last << endl;	
14		
15	strncpy(last,name,3);	
16	last[3]=0;	
17	cout << last << endl;	
18		
19	for (int i=0; i < strlen(name); i++)	
20	{	
21	cout << name[i] << " ";	
22	}	
23		
24	return 0;	
25	}	
26		

Задачи

1. Напишите C++ программу, считывающую два текста и число n, затем сравнивающую эти тексты с n т без него

2. Напишите C++ программу, считывающую два текста и создающую новый, состоящий из них

Тема : String class

В стандартной библиотеке (std), есть класс string. std::string позволяет работать в более упрощенном способе со string. Программист может присвоить string к ним, используя оператор присвоения, и тогда они автоматически будут увеличены или уменьшены в размерах, в зависимости от условий.
Для использования включите в C++ код #include <string>

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	Введите имя: Аскар
3	using namespace std;	Привет, Аскар
4		5
5	int main ()	
6	{	Попытка 2:
7	char init[] = "Введите имя :";	Введите имя:A B
8	stringname;	Привет, A
9	cout << init;	1
10	cin>>name;	
11		
12	cout << "Привет , " << name << endl;	
13		
14	cout << name.length();	
15		
16	return 0;	
17	}	

Задачи
1. Напишите C++ программу, считывающую имя и GPA двух студентов. Затем выведите имя студента с более высоким GPA.

Тема : Считывание string

Для чтения целых string используйте:
getline(cin, имя_ string);
Не забывайте что string может работать как массив.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	Введите имя: Аскар
3	using namespace std;	Привет Аскар
4		5
5	int main ()	Аск
6	{	
7	char init[] = "Введите имя :";	Попытка 2:
8	string name;	Введите имя: Аскар В
9	cout << init;	Привет А
10	getline(cin,name);	7
11		Аск
12	cout << "Привет " << name << endl;	
13		
14	cout << name.length() << endl;	
15	cout << name[0] << name[1] << name[2] << endl;	
16		
17	return 0;	
18	}	

Задачи
1. Напишите C++ программу, считывающую имя и фамилию в одной линии. Затем выведите их отдельно

2. Напишите C++ программу, считывающую арифметическое выражение как : “5 + 4 + 2” с однозначным числом и оператором + или - . Вычислите результат.

Тема : Конверсия

Последовательность символов, хранящаяся в char массивах может легко быть конвертирована в string объекты , используя оператор присвоения:

```
1  string str1;
2  char chr1[]="SDU";
3  str1 = chr1;
```

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	Введите имя:Askar
3	using namespace std;	Введите имя:Askar
4		
5	int main ()	Попытка 2:
6	{	Введите имя:A B
7	char init[] = "Введите имя :";	Введите имя:A B
8	string name, text;	
9	text = init;	
10		
11	cout << text;	
12	getline(cin,name);	
13		
14	cout << text << name << endl;	
15		
16	return 0;	
17	}	

Тема : Некоторые string функции

Существуют некоторые функции, облегчающие работу с string.

length	Длина string
clear	Удаление string
append	Добавление к string
push_back	Добавление символа к string
insert	Вставлять в string
replace	Замещать часть string
find	Находить содержимое в string

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	текст1
3		Текст2
4	using namespace std;	Текстт2
5		текстт1
6	int main ()	текст1
7	{	
8	string str1, str2;	
9		
10	cin >> str1 >> str2;	
11	swap(str1, str2);	
12	cout << str1 << str2 << endl;	
13		
14	\\\\	
15	str1.clear();	
16	cout << str1 << str2 << endl;	
17	return 0;	

Задачи

1. Напишите C++ программу, считывающую два string и затем производящая поиск буквы в обоих

Тема : Функция

В C++ структура функции:
тип_данных имя_функции (тип_данных имя_параметра) {}

Параметры могут быть использованы от 0 до необходимого количества. Параметры разделены запятыми(,). По параметру функция получает переменную(аргумент), который действует в функции как местная переменная.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	3 4
3		Площадь : 12
4	int area(int a, int b)	3.1
5	{	-3.1
6	int ar;	
7	ar = a*b;	Попытка 2:
8	return ar;	12 -4
9	}	Площадь : -48
10	double f1(double x)	-3.009
11	{	3.009
12	return -1*x;	
13	}	
14		
15	int main ()	
16	{	
17	int a1,a2;	
18	double d1,d2;	
19	cin>>a1>>a2;	
20	cout << "Площадь : " << area(a1,a2) << endl;	
21	cin>>d1;	
22	d2 = f1(d1);	
23	cout << d2;	
24		
25	return 0;	
26	}	

Задачи
1. Напишите функцию Макс, возвращающую максимум двух integer чисел

2. Напишите функцию Сумма, возвращающую сумму трех double чисел

Тема : Функции без типа “void”

В C++ программист может написать void type specifier (уточнение типа void) в типе_данных для функции. Структура :
void имя_функции(тип_данных имя_параметра) {}

Эти функции без типа используются, к примеру, для вывода значений массива, для вывода сообщения.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	Аскар
3	using namespace std;	Привет, Аскар
4		13
5	void area(string str)	26
6	{	
7	cout << "Привет, " << str << endl;	Попытка 2:
8	}	Арман
9	void show()	Привет, Арман
10	{	25
11	int a;	50
12	cin>>a;	
13	cout << a*2 << endl;	
14	}	
15		
16	int main ()	
17	{	
18	string str;	
19	cin>>str;	
20	area(str);	
21	show();	
22	return 0;	
23	}	

Задачи
1. Напишите C++ программу, считывающую 10 имен студентов с помощью цикла, затем напишите функцию print, получающую string и выводящая его с первой заглавной буквы

Тема : Среда в функции

Для следующей функции переменные являются местными.

```
void function1(int x, int y) // x и y создаются здесь
{
  int a=5;   // a создается здесь
} // x, y и a здесь удаляются
```

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	2 3
3		функция x : 2
4	int area(int x, int y)	функция y : 3
5	{	функция a : 6
6	int a;	----------
7	a = x*y;	функция x : 3
8	cout << "функция x : " << x << endl;	функция y : 4
9	cout << " функция y : " << y << endl;	функция a : 12
10	cout << " функция a : " << a << endl;	----------
11	cout << "----------" << endl;	главный x : 6
12	return a;	главный y : 12
13	}	главный a : 2
14		
15	int main ()	
16	{	
17	int a,b;	
18	cin>> a >> b;	
19	int x = area(a, b);	
20	int y = area(3, 4);	
21	cout << "главный x : " << x << endl;	
22	cout << " главный y : " << y << endl;	
23	cout << " главный a : " << a << endl;	
24	return 0;	
25	}	

Тема : Заблаговременная декларация

Прототип функции это декларация функции без ее реализации. Программист декларирует прототип функции в самом начале файла, чтобы показать компилятору что она будет использоваться. Это называется заблаговременная декларация.
Если заблаговременная декларация функции не используется в коде и функция написана после main функции, тогда при попытке запуска программы компилятор выдает ошибку.

int main()
{
int a = sum(3,4);
}
 	
int sum(int x, int y)
{
return x+y;
}
	В коде слева есть компиляционная
	ошибка
	Для решения этой ошибки:
	int sum(int, int);
	
	int main()
	{
	…. То же что и слева

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	5 6
3	intarea(int,int); //заблаговременная декларация, попытка сделать	Площадь: 30
4	// это комментарием	
5	//чтобы понять и увидеть ошибку	Попытка 2:
6		12 2
7	int main ()	Площадь: 24
8	{	
9	int a,b;	
10	cin>> a >> b;	
11	int x = area(a, b);	
12	cout << "Площадь : " << x << endl;	
13	return 0;	
14	}	
15		
16	int area(int x, int y)	
17	{	
18	return x*y;	
19	}	

Задачи
1. Напишите функции площадь, периметр круга, используя заблаговременную декларацию

Тема : Внешняя функция

В C++ программист может писать функции в других файлах и использовать их. Для использования внешнего файла в C++ коде он должен быть включен в файл и иметь расширение .h. Структура:
#include "имя_файла.h"
Замечание: есть разница между <> и" "в C++. В стандартном классе включено <>, в собственном местном классе :" ".

	C++ КОД	РЕЗУЛЬТАТ
	sum.h	Попытка 1: 3 4 Площадь: 12 Попытка 2: 6 9 Площадь: 54

1
2
3
4
	int area(int x, int y)
	{
	return x*y;
	}	

	main.cpp	

1
2
3
4
5
6
7
8
9
10
11
12
	#include <iostream>
	#include "sum.h" // здесь включена внешняя
	//функция
	using namespace std;
	
	int main ()
	{
	int a,b;
	cin>> a >> b;
	int x = area(a, b);
	cout << "Площадь : " << x << endl;
	return 0;	

Задачи
1. Напишите две функции, которые находят максимум и минимум 3 integer чисел. Сохраните оба в файле test.h и используйте в main.cpp

Тема : Ключевое слово Namespace

Когда два identifier (имя переменной и/или функции) с одинаковым именем вводятся в одинаковую среду, происходит именное столкновение и компилятор создает ошибку, т.к. у него недостаточно информации для устранения двусмысленности.

Ключевое слово namespace определяет место в коде , где все identifier(имена) гарантированы быть уникальными. По умолчанию, все переменные и функции определены в глобальном namespace. Во избежание случаев, где две независимые части кода имеют именные столкновения в процессе одновременного использования, C++ позволяет нам декларировать наши собственные namespace(названия) с помощью кодового слова namespace. Все, что было декларировано в обозначенном пользователем namespace принадлежит ему, а не глобальному namespace.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: 5 3 namespace n1 : 8 namespace n2 : 2 Попытка 2: -3 4 namespace n1 : 1 namespace n2 : -7
2	using namespace std;	
3	namespace n1	
4	{	
5	int function(int x, int y)	
6	{	
7	return x + y;	
8	}	
9	}	
10	namespace n2	
11	{	
12	int function(int x, int y)	
13	{	
14	return x - y;	
15	}	
16	}	
17		
18	int main ()	
19	{	
20	int a,b;	
21	cin>> a >> b;	
22	intx1=n1::function(a,b);	
23	// или использовать namespace n1	
24	cout << "namespace n1 : " << x1 << endl;	
25	intx2=n2::function(a,b);	
26	// или использовать namespace n2	
27	cout << "namespace n2 : " << x2 << endl;	
28		
29	return 0;	
30	}	

Задачи

1. Напишите два namespaces: экономика и инженерия. В обоих есть переменная цена и функция вычисление, которая возвращает 500 и 700 экономики и инженерии соответственно

Тема : Проходя аргументы

Есть три главных метода, чтобы пройти через аргументы: через значение, через ссылку, через адрес

1. Проходя аргументы через значение. По умолчанию, аргументы в C++ проходят через значение. Когда аргументы проходят через значение, в функцию передается копия аргумента.

2. Проходя через ссылку. Когда аргументы проходят через ссылку, программист может менять значение аргумента. Для прохождения через ссылку используется знак адреса (&).
void function(int &y) // y это переменная ссылки

3. Проходя через адрес. Это означает прохождение через адрес аргумента переменной быстрее чем аргумент себя переменит. Так как аргумент это адрес, функция параметра должна быть указателем. Функция может снять ссылку с указателя для доступа или изменения указанного значения.
void function(int *y) // y это указатель
function(&a); // в main программист отсылает указатель к переменной, а не к ее значению.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	3 5
3	void f1(int, int &);	3 5
4	void f2(int, int *);	3 5
5	int main ()	6 10
6	{	3 10
7	int a,b;	---------
8	cin>> a >> b;	3 10
9	cout << a << " " << b << endl;	3 10
10	f1(a,b);	9 30
11	cout << a << " " << b << endl;	3 30
12	cout << "----------" << endl;	
13	cout << a << " " << b << endl;	
14	f2(a,&b);	
15	cout << a << " " << b << endl;	
16	return 0;	
17	}	
18	void f1(int a, int &b)	
19	{	
20	cout << a << " " << b << endl;	
21	a = a*2;	
22	b = b*2;	
23	cout << a << " " << b << endl;	
24	}	
25	void f2(int a, int *b)	
26	{	
27	cout << a << " " << *b << endl;	
28	a = a*3;	
29	*b = (*b) * 3;	
30	cout << a << " " << *b << endl;	
31	}	

Задачи

1. Напишите две функции, меняющие местами два integer числа ссылкой, адресом

Тема : Класс

Класс это расширенная концепция структуры данных: вместо содержания только данных, он содержит и данные, и функции

Объект это инстантация класса. В отношении переменных , класс будет типом , а объект это переменная.
Структура :
class имя_класса{
переменные и функции
};

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	Арман 25
3	using namespace std;	Имя: Арман
4	class person // описание класса	Возраст: 25
5	{	
6	public :	Попытка 2:
7	string name; // переменная	Андрей 19
8	int age; // переменная	Имя : Андрей
9	void print() // функция	Возраст: 19
10	{	
11	cout << "Имя : " << name << endl;	
12	cout << "Возраст : " << age << endl;	
13	}	
14	}; // важно использовать знак ';'	
15	int main ()	
16	{	
17	person student; //объект класса person	
18	string n;	
19	int a;	
20	cin>> n >> a;	
21	student.name = n;	
22	student.age = a;	
23	student.print();	
24	return 0;	
25	}	
26		

Задачи
1. Напишите класс дата, имеющий 3 переменные: год, месяц и день. Напишите функцию print , которая будет выводить дату в формате: “день месяца, год”

2. Напишите класс книга, имеющий 2 переменные: имя, автор. Напишите функцию show, которая выводит книги в формате“имя автора”

Тема : Уровень доступа

Уровень доступа это декларация для данных в классе, для предоставления доступа. Уровень доступа представлен тремя ключевыми словами: private(приватный), public(публичный) и protected(защищенный).

- public члены доступны везде, где видим объект.
- private члены доступны только для остальных членов того же класса или их дружественных классов.
- protected члены доступны не только для членов того же класса и их дружественных классов , но и для членов их полученных классов (дочерние классы).

Замечание: по умолчание вся информация в классе private типа.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	Аскар 25
3	using namespace std;	Имя: Аскар
4	class person	Возраст: 25
5	{	
6	private : // переменные name и age private	Попытка 2:
7	string name;	Арман 33
8	int age;	Имя: Арман
9	public : // функции public	Возраст: 33
10	void set(string a, int b)	
11	{	
12	name = a;	
13	age = b;	
14	}	
15	void print()	
16	{	
17	cout << "Имя : " << name << endl;	
18	cout << "Возраст : " << age << endl;	
19	}	
20	};	
21		
22	int main ()	
23	{	
24	personstudent;	
25	string n;	
26	int a;	
27	cin>> n >> a;	
28	student.set(n,a); //работает, т.к. public	
29		
30	//student.name = n; // без комментария это КО	
31	//student.age = a; т.к. private	
32		
33	student.print();//опять public	
34	return 0;	
35	}	

*КО- компиляционная ошибка

Задачи

1. Напишите класс student , где есть две public функции setInfo и getInfo и 3 private переменные имя, GPA, курс

Тема : Функция доступа

В классе может быть прототип декларации функции, но ее реализация может находиться вне класса. В случае нахождения вне класса программист должен использовать оператор среды (::) , для уточнения того, что функция является членом этого класса и того что она не является обычной глобальной функцией.

В классе программисты используют оператор среды (::, два двоеточия) для обозначения члена класса , находящегося за пределами класса .

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	Аскар 3 3.8
3	using namespace std;	Имя : Аскар
4	class student	Курс : 3
5	{	GPA : 3.8
6	string name;	----------
7	double GPA;	Андрей1 2.5
8	int course;	Имя : Андрей1
9	public :	Курс : 1
10	void set(string, int, double); // только	GPA : 2.5
11	//прототипы	
12	void print()	
13	{	
14	cout << "Имя : " << name << endl;	
15	cout << "Курс : " << course << endl;	
16	cout << "GPA : " << GPA << endl;	
17	}	
18	};	
19	void student::set(string a, int b, double c) // реализация	
20	{	
21	name = a;	
22	course = b;	
23	GPA = c;	
24	}	
25	int main ()	
26	{	
27	student s1, s2;	
28	string n;	
29	int a;	
30	double d;	
31	cin>> n >> a >> d;	
32	s1.set(n, a, d);	
33	s1.print();	
34	cout << "----------"<<endl;	
35	cin>> n >> a >> d;	
36	s2.set(n, a, d);	
37	s2.print();	
38	return 0;	

Задачи

1. Напишите класс Rectangle с private переменной ширина и длина, с функциями setWidth(установить ширину), setLength(установить длину), Area(площадь). Все реализации

функции напишите вне класса.

Тема : Конструктор

Конструктор это специальная функция, используемая для инициализации переменных в классе. Он автоматически вызывается вне зависимости от того создан ли новый объект этого класса или нет. Эта функция конструктора должна иметь одинаковое имя с классом и не иметь никакого типа возврата, даже не void.

Структура :

имя_класса(тип_данных parameter)

{ C++ код }

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string> using	Аскар 2 3.5
3	namespace std; class	Создан новый студент
4	student	Имя : Аскар
5	{	Курс : 2
6	string name;	GPA : 3.5
7	double GPA;	
8	int course;	Попытка 2:
9	public :	Олжас 1 3.8
10	student(string, int, double); //конструктор	Создан новый студент
11	void print();	имя : Олжас
12	};	Курс : 1
13	void student::print()	GPA : 3.8
14	{	
15	cout << "Имя : " << name << endl;	
16	cout << "Курс : " << course << endl;	
17	cout << "GPA : " << GPA << endl;	
18	}	
19	student::student(string a, int b, double c) // реализация	
20	{	
21	cout << "Создан новый студент " << endl;	
22	name = a;	
23	course = b;	
24	GPA = c;	
25	}	
26		
27	int main ()	
28	{	
29	string n; int a;	
30	double d;	
31	cin>> n >> a >> d;	
32	students1(n,a,d);	
33	s1.print();	
34		
35	return 0;	
36	}	
37		

Задачи

1. Напишите класс time1 с конструктором (int hour, int minute, int seconds).(час,минута,секунда)

2. Напишите класс notebook с конструктором (string model, int year, string company) (модель, год, компания)

Тема : Перегружая конструктор

В классе может быть один и более конструкторов с разными типами или количеством параметров. Это называется перегружением конструктора Все они должны иметь одинаковое имя, но компилятор будет вызывать только те конструкторы, чьи параметры совпадают с аргументами.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	3 5
3	class rectangle	Без параметров
4	{	Создан квадрат
5	int width, height;	Создан прямоугольник
6	public :	9 15
7	rectangle(); // без параметров	
8	rectangle(int); // с параметром width.Square	Попытка 2:
9	rectangle(int, int); // с параметрами width и height	4 7
10	int area();	Без параметров
11	};	Создан квадрат
12	rectangle::rectangle() {	Создан прямоугольник
13	width = height = 0;	16 28
14	cout << "Без параметров " << endl; }	
15	rectangle::rectangle(int w) {	
16	width = w;	
17	height = w;	
18	cout << "Создан квадрат " << endl; }	
19	rectangle::rectangle(int w, int h) {	
20	width = w;	
21	height = h;	
22	cout << " Создан прямоугольник" << endl; }	
23	int rectangle::area() {	
24	return width*height;}	
25		
26	int main ()	
27	{	
28	int a,b;	
29	cin >> a >> b;	
30	rectangle r1; // constructor1	
31	rectangle r2(a); // constructor2	
32	rectangle r3(a, b); // constructor3	
33		
34	cout << r2.area() << " " << r3.area();	
35	return 0;	
36	}	

Задачи

1. Напишите класс cellPhone с переменными: manufacture(завод), model(модель), price(цена) и color(цвет). Напишите 2 перегруженных конструктора для этого класса. Напишите функции changePrice(изменить цену) и changeColor(изменить цвет).

1. Напишите класс book с переменными name(имя), author(автор), pages(страницы) и ISBN.

Напишите 3 перегруженных конструктора для этого класса. Напишите функцию changeName(изменить имя) и print(вывод на экран)

Тема : Стандартный конструктор

Если в в дефиниции класса отсутствует декларация конструктора, компилятор предполагает что класс имеет стандартный конструктор без аргументов. Программист может декларировать объекты класса без аргументов. Но так как программист декларирует собственный конструктор для класса, компилятор больше не обеспечивает implicit стандартный конструктор. Поэтому программисту следует декларировать все объекты класса, соответствуя прототипу конструктора.

class example1
{
public :
int a;
void setA(int b) { a= b; }
};
	class example2
	{
	public :
	int a;
	example2(int b) { a= b; }
	};

Без конструктора, поэтому мы можем писать: example1 d; И не можем писать: example1 d(5);	С конструктором, поэтому можем писать example2 d(5); И не можем писать: example2 d;

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: 3 3 3 Попытка 2: 4 4 4
2	using namespace std;	
3	class example1	
4	{	
5	public :	
6	int a;	
7	void setA(int b) { a= b; }	
8	};	
9	class example2	
10	{	
11	public :	
12	int a;	
13	example2(int b) { a= b; }	
14	};	
15	int main ()	
16	{	
17	int x;	
18	cin >> x;	
19	example1 r1;	
20	r1.setA(x);	
21		
22	example2r2(x);	
23		
24	cout << r1.a << " " << r2.a;	
25	return 0;	
26	}	

Задачи

1. Напишите класс «инженерная группа» с конструктором и «экономическая группа»без конструктора с функцией print(вывод).Напишите main class для их использования

Тема : Деструктор

Деструктор это обратное конструктора. Обычно он вызывается автоматически, когда уничтожается объект, так как завершилась его среда существования или потому что объект динамически приравнен и освобожден, используя оператор delete. Или можно сделать свой собственный деструктор.

Структура :

```
~имя_класса ()
{
}
```

Замечание : в классе может быть только один деструктор.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	#include <string>	3 5
3	using namespace std;	3 5
4	class book	3 5
5	{	6 10
6	string *name;	3 10
7	int *ISBN;	---------
8	public :	3 10
9	book(string n, int i); // конструктор	3 10
10	~book(); // деструктор	9 30
11	};	3 30
12	book::book(string n, int i)	
13	{	
14	name = new string(n);	
15	ISBN = new int;	
16	*ISBN = i;	
17	cout << "Объект " << n << endl;	
18	}	
19	book::~book()	
20	{	
21	cout << "Объект уничтожен" << endl;	
22	delete ISBN;	
23	delete name;	
24	}	
25	int main ()	
26	{	
27	string s;	
28	int x;	
29	cin >> s >> x;	
30	{	
31	book b1(s, x);	
32	} // здесь работает деструктор	
33	book b2("AAA",x);	
34	return 0;	
35	} // здесь работает деструктор	

Задачи

1. Напишите класс student с деструктором

Тема : Внешний класс

Программист может обозначить и реализовать класс для использования других ресурсов.
Чаще всего в (.h) головном файле есть дефиниция класса. И в (.cpp) есть реализация этого класса.

example.h

```
1  class example
2  {
3          int i;
4          double d;
5  public :
6          example();
7          void set(int, double);
8          void print();
9  };
```

example.cpp

```
1  #include <iostream>
2  #include "example.h"
3  using namespace std;
4
5  example::example() { }
6  void example::set(int a, double q)
7  {
8          i = a;
9          d = q;
10 }
11 void example::print()
12 {
13         cout << "Integer " << i << endl;
14         cout << "Double " << d << endl;
15 }
```

main.cpp

```
1  #include <iostream>
2  #include "example.h"
3  using namespace std;
4
5  int main()
6  {
7    int x;
8    double d;
9    cin >> x >> d;
10   example s1;
11   s1.set(x, d);
12   s1.print();
13   return 0;
14 }
```

Результат

```
Попытка 1:
2 3.1
Integer 3
Double 3.1

Попытка 2:
4 4.11
Integer 4
Double 4.11
```

Задачи

1. Напишите C++ класс «book»(книга) с (.h) файлом и реализацией класса. Напишите main класс для тестирования

Тема : Указатель к классу

В C++ программист может сделать указатель к классу. Структура:
имя_класса *имя_объекта;
Есть два способа вызова функции:
1 (*имя_объекта).method();
2 – <имя_объекта >method();

	C++ Код	Результат
1	#include <iostream>	Попытка 1:
2	#include <string>	Имя Неизвестно
3	using namespace std;	Автор Неизвестно
4	class Book	Абай
5	{	Көзімнің қарасы
6	string name, author;	Имя Абай жолы
7	public :	Автор Ауезов
8	Book();	Имя Абай Автор
9	void setBook(string, string);	Автор Көзімнің Имя
10	void print();	Неизвестно Автор
11	};	Неизвестно Имя
12	Book::Book() {	Неизвестно Автор
13	name = "Неизвестно";	Неизвестно
14	author = " Неизвестно "; }	
15	void Book::setBook(string n, string a) {	Попытка 2:
16	name = n;	Имя Неизвестно
17	author = a; }	Автор Неизвестно
18	void Book::print() {	Оқұлық
19	cout << "Имя " << name << endl;	Алтынсарин
20	cout << "Автор " << author << endl;}	Имя Абай жолы
21		Автор Ауезов
22	int main ()	Имя Оқұлық
23	{	Автор Алтынсарин
24	Book r1;	Имя Неизвестно
25	r1.print();	Автор Неизвестно
26	string a,b;	Имя Неизвестно
27	cin >> a >> b;	Автор Неизвестно
28	Book *r2 = new Book[2];	
29	r2->setBook("Абай жолы", "Ауэзов");	
30	r2[0].print();	
31	r2[1].setBook(a,b);	
32	r2[1].print();	
33		
34	Book *r3=&r1;	
35	r3->print();	
36	(*r3).print();	
37	return 0;	
38	}	

Задачи
1. Напишите C++ класс mobile phone с указателем к классу

Тема : Указатель “this”

Ключевое слово «this» определяет специальный тип указателя. Оно используется для хранения адреса вызываемого объекта. Предположим , что вы создали объект под названием «x» класса «A» и класс «A» имеет функцию f() типа nonstatic.Если вы вызываете функцию x.f(), ключевое слово «this» в главной части функции f() содержит адрес x. Нельзя декларировать указатель «this»или ставить для него задачи.

	C++ КОД	Результат
1	#include <iostream>	Попытка 1:
2	using namespace std;	2
3	class calc	2
4	{	0
5	int a;	15
6	public :	
7	calc();	Попытка 2:
8	void set(int);	4
9	int get();	4
10	calc& plus(int);	0
11	};	15
12	calc::calc() {	
13	a = 0; }	
14	void calc::set(int x) {	
15	a = x; }	
16	int calc::get(){	
17	return a;	
1	}	
19	calc& calc::plus(int x) {	
20	a += x;	
21	return *this; }	
22		
23	int main ()	
24	{	
25	calc c1;	
26	calc c2;	
27	int x;	
28	cin >> x;	
29	c1.set(x);	
30	cout << c1.get() << endl;	
31	cout << c2.get() << endl;	
32	c2.plus(10).plus(5);	
33	cout << c2.get() << endl;	
34		
35	return 0;	
36	}	

Задачи

1. Напишите C++ класс number с объектом n1 и указателем к нему. Создайте функцию check, которая проверяет совпадает ли указатель с объектом.

Тема : Операторы перегружения

Чтобы прегрузить оператор для использования с классами программисты декларируют функции оператора, являющимися регулярными функциями . Их имена обозначаются ключевым словом operator keyword со знаком оператора. Структура: тип_данных знак оператора(параметры)
{ C++ КОД }

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	1 2
3		3 3
4	class number {	
5	public:	Попытка 2:
6	int x;	3 7
7	number () {};	10 10
8	number (int);	
9	number operator+(number);	
10	};	
11	number::number(int a) {	
12	x = a;	
13	}	
14	number number::operator+(number n1) {	
15	number temp;	
16	temp.x = x + n1.x;	
17	return temp;	
18	// return number(x+n1.x);(возвращает число(x+n1.x))	
19	}	
20		
21	int main () {	
22	int a1, a2;	
23	cin >> a1 >> a2;	
24	number a(a1);	
25	number b(a2);	
26	number c, d;	
27	c = a + b;	
28	d = a.operator+(b);	
29	cout << c.x << " " <<d.x <<endl;	
30	return 0;	
	}	

Задачи
1. Напишите класс number , вычитающий два числа.

Тема : Перегружение единичного оператора

Единичный оператор во всех случаях использования работает с одним операндом. Поэтому операнд становится вызывающим с использованием типа данных, установленных пользователем. Следовательно, аргументы не нужны.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	1 2
3		2 3
4	class number {	
5	public:	Попытка 2:
6	int x;	34 45
7	number () {};	35 46
8	number (int);	
9	number operator ++();	
10	number operator ++ (int);	
11	};	
12	number::number(int a) {	
13	x = a;	
14	}	
15	number number::operator++()	
16	{	
17	return number(++x);	
18	}	
19	number number::operator++(int)	
20	{	
21	return number(x++);	
22	}	
23		
24	int main () {	
25	int a1, a2;	
26	cin >> a1 >> a2;	
27	number a(a1);	
28	number b(a2);	
29	a++;	
30	++b;	
31	cout << a.x << " " <<b.x <<endl;	
32	return 0;	
33	}	

Задачи

1. Напишите класс number декремент оператором.

Тема : Перегружение бинарного оператора

Бинарный оператор во всех случаях использования работает с двумя операндами. Поэтому первый операнд становится оператором, перегрузившим функцию вызова, а второй проходит как аргумент. Используется тип данных, установленных пользователем. Результатом является принуждение в получении одного аргумента в перегружении бинарных операторов

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	2 3
3		Не равны
4	class number {	
5	int x;	Попытка 2:
6	public:	3 3
7	number () {};	Равны
8	number (int);	
9	bool isEqual(number);	
10	};	
11	number::number(int a) {	
12	x = a;	
13	}	
14	bool number::operator==(number n1)	
15	{	
16	return (x == n1.x);	
17	}	
18		
19	int main () {	
20	int a1, a2;	
21	cin >> a1 >> a2;	
22	number a(a1);	
23	number b(a2);	
24	if(a==b) cout<<"Равны"<<endl;	
25	else cout << " Не равны";	
26	return 0;	
27	}	

Задачи

1. Напишите класс number, проверяющий больше, меньше или равны ли числа

Тема : перегружение ввода/вывода оператора

В C++ есть функции перегружения cin и cout. Это означает что программист может создать функцию класса для чтения данных именно так, как ему нужно

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	3 5
3		x : 3 y : 5
4	class Cnumber {	4 5
5	int x, y;	x : 4 y : 5
6	public:	
7	Cnumber () {};	Попытка 2:
8	Cnumber (int, int);	5 5
9	friend istream& operator>> (istream &in, Cnumber&t);	x : 5 y : 5
10	friend ostream& operator<< (ostream &out, Cnumber&t);	12 13
11	};	x : 12 y : 13
12	Cnumber::Cnumber(int a, int b) {	
13	x = a;	
14	y = b;	
15	}	
16	istream& operator>>(istream &in, Cnumber &t)	
17	{	
18	in >> t.x;	
19	in >> t.y;	
20	return in;	
21	}	
22	ostream& operator<< (ostream &out, Cnumber &t)	
23	{	
24	out << " x : " << t.x << " y : " << t.y << endl;	
25	return out;	
26	}	
27		
28	int main () {	
29	Cnumber a1, a2, a3;	
30	cin >> a1;	
31	cout << a1;	
32	cin >> a2;	
33	cout << a2;	
34		
35	return 0;	
36	}	
37		

Задачи

1. Напишите класс number с перегружением оператора ввода/вывода

2. Напишите класс time1 с переменными int hour(час), int minute(минута), int seconds(секунды). Сделайте перегружение оператора ввода/вывода.

Тема : Дружественная функция

Дружественная функция это функция, которая имеет доступ к private частям класса, как если бы она сама была членом класса. В остальном дружественная функция является обычной

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	3 5
3		Без параметров
4	class number {	Создан квадрат
5	int x;	Создан прямоугольник
6	public:	9 15
7	number () {};	
8	number (int);	Попытка 2:
9	friend number operator+(number, number);	4 7
10	void print() { cout << x;}	Создан квадрат
11	};	Создан прямоугольник
12	number::number(int a) {	16 28
13	x = a;	
14	}	
15	number operator+(number t1, number t2)	
16	{	
17	number temp;	
18	temp.x = t1.x + t2.x;	
19	return temp;	
20	}	
21	int main () {	
22	int a1, a2;	
23	cin >>a1;	
24	numbera(a1);	
25	cin >> a2;	
26	numberb(a2);	
27		
28	numberc;	
29	c = a + b;	
30	c.print();	
31	return 0;	
32	}	

Задачи

1. Напишите дружественный оператор для члена класса с двумя аргументами: первый с объектом, второй с integer числом.

Тема : Дружественный класс

Дружественный класс обозначает класс, являющийся другом для другого. Это дает доступ первому классу к protected и private (защищенным и приватным) членам второго.

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	3
3		9
4	class CSquare;	
5		Попытка 2:
6	class CRectangle {	4
7	int width, height;	16
8	public:	
9	int area ()	
10	{return (width * height);}	
11	void convert (CSquare a);	
12	};	
13	void CRectangle::convert (CSquare a) {	
14	width = a.side;	
15	height = a.side;	
16	}	
17		
18	class CSquare {	
19	private:	
20	int side;	
21	public:	
22	void set_side (int a) {side=a;}	
23	friend class CRectangle;	
24	};	
25		
26	int main () {	
27	CSquare sqr;	
28	CRectangle rect;	
29	int a;	
30	cin >> a;	
31	sqr.set_side(a);	
32	rect.convert(sqr);	
33	cout << rect.area();	
34	return 0;	
35	}	
36		

Тема : Наследование

Наследование это способ создания классов, полученных от других классов. Отсюда следует, что они автоматически включают в себя некоторых «родительских» членов и собственных. Класс, от которого унаследовали что-либо, называется родительским классом или основой, а класс который что-либо унаследовал, называется дочерним или полученным объектом.

	C++ КОД	РЕЗУЛЬТАТ

```
#include <iostream> using
namespace std; class
person{
public:
   string name;
   int age;
   void setName(string s) { name = s; }
   void setAge(int i){ age = i; } string
   getName() { return name; } int
   getAge() { return age; }
   void print() {
        cout << "Имя:" << name << " Возраст:" << age << endl;}
};
class student:public person{
public:
   double GPA;
   int course;
   void setGPA(double d) { GPA = d; } void
   setCourse(int i) { course = i; } void info()
   {
          print();
          cout << "GPA:" << GPA<< " Курс:" << course <<
endl;
           }
};
int main () {
 students1;
  string name;
  cin >> name;
 s1.setName(name);
  int age;
  cin >> age;
  s1.setAge(age);
  s1.print(); double g;
  int c;
  cin >> g >> c;
  s1.setGPA(g);
  s1.setCourse(c);
  s1.info();
  return 0;
}
```

```
Попытка 1: Askar 21
Имя:Askar
Возраст:21
3.4
2
Имя:Askar
Возраст:21
GPA:3.4
Курс:2

Попытка 2:
Arman 19
Имя:Arman
 Возраст:19
2.6
1
Имя:Arman
Возраст:19
GPA:2.6
 Курс:1
```

Задачи

1. Напишите класс shape(размер) с переменными width(ширина) и height(высота).Напишите дочерний класс rectangle(прямоугольник) с функцией area(площадь).

Тема : Защищенные члены

Защищенный уровень доступа схож с приватным. Единственная разница в наследовании состоит в том, что когда класс наследуется от другого, члены полученного класса имеют доступ только к защищенным членам , унаследованных от основного класса, но не к приватным .

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	Аскар Аскаров
3	class person{	2.3
4	private:	1
5	string surname;	Имя студента:Аскар
6	protected: string name;	GPA:2.3
7	int age;	Курс:1
8	public:	
9	void setName(string s) { name = s; }	Попытка 2: Арман
10	void setAge(int i){ age = i; }	Амангелды
11	void setSurname(string s) { surname = s; }	3.6
12	void print() {	3
13	cout << "Имя:" << name << " Возраст:" << age << endl;}};	Имя студента:Арман
14	class student:public person{	GPA:3.6
15	public:	Курс:3
16	double GPA;	
17	int course;	
18	void setGPA(double d) { GPA = d; }	
19	void setCourse(int i) { course = i; } void	
20	info() {	
21	cout << "Имя студента:" << name << endl;	
22	// cout << "Фамилия студента:" << surname << endl;	
23	// эта строка не может быть использована, т.к.	
24	//она приватная	
25	cout << "GPA:" << GPA<< " Курс:" << course	
26	<< endl;	
27	}	
28	};	
29	int main () {	
30	students1;	
31	string name, surname;	
32	cin >> name;	
33	s1.setName(name);	
34	cin >> surname;	
35	s1.setSurname(surname);	
36	double g;	
37	int c;	
38	cin >> g >> c;	
39	s1.setGPA(g);	
40	s1.setCourse(c);	
41	s1.info();	
42	return 0;	
43	}	

Задачи

1. Напишите класс point(точка) с защищенным int x, y. Напишите дочерний класс line(линия)

Тема : Связь наследования

Есть три способа наследования: public, private, and protected.

class Child: public Parent {};	Все члены сохраняют исходный уровень доступа. Private остаются private, protected члены остаются protected и public members остаются public.
class Child: private Parent {};	private члены остаются private, a protected и public становятся private.
class Child: protected Parent {};	public и protected члены становятся protected, a private члены остаются private
class Child: Parent {};	По умолчанию используется private

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1: Компиляционная ошибка Потому что кваларт пытается использовать ширину и высоту фигуры . Они являются private
2	using namespace std;	
3	class shape{	
4	private:	
5	int Z;	
6	protected:	
7	int width, height;	
8	public:	
9	void set(int a, int b)	
10	{	
11	width = a;	
12	height = b;	
13	}	
14	};	
15	class rectangle:private shape{	
16	public:	
17	int getArea()	
18	{	
19	return width*height;	
20	}	
21	};	
22	class square:public rectangle{	
23	public:	
24	int getArea()	
25	{	
26	return width*height;	
27	}	
	};	
	int main () {	
	rectangle r;	
	square s;	
	return 0;	
	}	

Тема : Инициализация полученных классов

Полученный класс наследует каждого члена основного класса, кроме:

- его конструктор и деструктор
- его члены оператора =()
- его дружественных классов

Для перегружения конструктора основного класса новый полученный объект создается структурой:
полученный_ конструктор _имя (параметры) : основной_ конструктор _имя (параметры)
{ C++ код }

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	5
3	class A	3.4
4	{	X
5	public:	A: 5
6	A(int n)	B: 3.4
7	{	C: X
8	cout << "A: " << n << endl;	
9	}	Попытка 2:
10	};	9
11	class B: public A	4.1
12	{	o
13	public:	A: 9
14	B(int n, double d) : A(n)	B: 4.1
15	{	C: o
16	cout << "B: " << d << endl;	
17	}	
18	};	
19	class C: public B	
20	{	
21	public:	
22	C(int n, double d, char ch)	
23	: B(n, d)	
24	{	
25	cout << "C: " << ch << endl;	
26	}	
27	};	
28	int main()	
29	{	
30	int i1;	
31	double i2;	
32	char i3;	
33	cin >> i1 >> i2 >> i3;	
34	C c1(i1, i2, i3);	
35	return 0;	
36	}	

Тема : Множественное наследование

Множественное наследование включает полученный класс для наследования членов от более чем одного родителя. Структура:

class имя_полученного_класса : public родитель1_класс, public родитель2_класс e.t.c

{ C++ код };

	C++ КОД	РЕЗУЛЬТАТ
1	#include <iostream>	Попытка 1:
2	using namespace std;	Арман
3	class person	87019999999
4	{	2
5	protected:	Имя студента: Арман
6	string name;	Номер телефона:
7	public:	870199999999
8	void setName(string n) { name = n; }	Курс: 2
9	};	
10	class mobilePhone	Попытка 2:
11	{	Аскар
12	protected:	2738800
13	string phone;	1
14	public:	Имя студента: Аскар
15	void setPhone(string n) { phone = n;}	Номер телефона: 2738800
16	};	Курс: 1
17	class student: public person, public mobilePhone	
18	{	
19	int course;	
20	public:	
21	void info() {	
22	cout << "Имя студента : " << name << endl;	
23	cout << "Номер телефона : " << phone <<	
24	endl; cout << "Курс : " << course << endl;	
25	}	
26	void setCourse(int i) { course = i; }	
27	};	
28	int main()	
29	{	
30	string n, p;	
31	student s1;	
32	cin >> n;	
33	s1.setName(n);	
34	cin >> p;	
35	s1.setPhone(p);	
36	int i;	
37	cin >> i;	
38	s1.setCourse(i);	
39	s1.info();	
40	return 0;	
41	}	

Задачи

1. Напишите класс child, с 2 родительскими классами father и mother.

Ссылки

1. www.cplusplus.com
2. www.learncpp.com
3. http://www.codingunit.com
4. http://www.mochima.com/tutorials