**Лабораторна робота №9**

**Виконав студент групи ТІ-92 1-го курсу**

**П.І.Б Черноусов Денис Ігорович**

Завдання :

Задача 9.1. Создайте иерархию классов для хранения информации о летательных аппаратах. Начните с общего базового класса Airship, предназначенного для хранения количества перевозимых пассажиров и количества перевозимого груза в фунтах или килограммах, или тоннах. Затем создайте два производных от Airship класса Airplane и Balloon (дирижабль). Класс Airplane должен хранить тип самолетного двигателя (винтовой или реактивный) и дальность полета в милях или километрах. Класс Balloon должен хранить информацию о типе газа, используемого для подъема дирижабля (водород или гелий), и его максимальный потолок (в футах или метрах). Создайте программу для демонстрации работы этой иерархии классов.

Задача 9.2. Напишите программу, которая бы устанавливала флаги для потока cout так, чтобы целые, если они положительны, выводились со знаком +. Покажите, что ваш набор флагов формата правилен.

Задача 9.3. Напишите программу, которая бы устанавливала флаги для потока cout так, чтобы всегда при выводе дробных значений были показаны десятичные точки. Кроме этого, значения с плавающей точкой должны выводиться в научной нотации с символом "Е" в верхнем регистре.

Задача 9.4. Напишите программу, которая сохраняет текущее состояние флагов формата, устанавливает флаги showbase и hex, выводит на экран значение 100, а затем возвращает флаги в исходное состояние.

Задача 9.5. Разработайте программу для печати таблицы натуральных и десятичных логарифмов чисел от 2 до 100. Формат таблицы следующий: правое выравнивание, ширина поля — 10 символов, точность — 5 десятичных позиций.

Задача 9.6. Создайте функцию center() со следующим прототипом: void center(char \*s); Эта функция должна устанавливать заданную строку в центр экрана. Для реализации этой задачи воспользуйтесь функцией width(). Предполагается, что ширина экрана равна 80 символам. (Для простоты считайте, что длина строки не превышает 80 символов.) Напишите программу, иллюстрирующую работу этой функции. Совет: Поэкспериментируйте с флагами и функциями формата. После того как вы ближе познакомитесь с системой ввода/вывода C++, вы, хочется верить, никогда не ошибетесь в выборе нужного формата вывода. :)

Задача 9.7. Выполните еще раз Задачу 9.5 и 9.6, только теперь, вместо функций-членов и флагов формата, используйте манипуляторы ввода/вывода.

Задача 9.8. Составьте инструкцию для вывода числа 100 в шестнадцатеричной системе счисления с отображением основания системы счисления (0x). Для выполнения задания воспользуйтесь манипулятором setiosflags().

Задача 9.9. В незавершенной программе имеется класс strtype. Для вывода строки на экран создайте пользовательскую функцию вывода: #include #include #include using namespace std; class strtype { char \*p; int len; public: strtype(char \*ptr); ~strtype() { delete [] p; } friend ostream &operator <<< "Allocation error\n"; exit(1); } strcpy(p, ptr); } // Create operator << inserter function here. int main() { strtype s1("This is a test."), s2("I hate C++!"); cout << s1 << '\n' << s2; return 0; }

Задача 9.10. Замените в следующей программе функцию show() пользовательской функцией вывода: #include using namespace std; class planet { protected: double distance; // расстояние в милях от Солнца int revolve; // полный оборот в днях public: planet(double d, int r) { distance = d; revolve = r; } }; class earth : public planet { double circumference; // окружность орбиты public: earth(double d, int r) : planet(d, r) { circumference = 2 \* distance \* 3.1416; } /\* Rewrite this so that it displays the information using an inserter function. \*/ void show() { cout << "Distance from sum: " << distance << '\n'; cout << "Days in orbit: " << revolve << '\n'; cout << "Circumference of orbit: " << circumference << '\n'; } }; int main() { earth obj(93000000, 365); cout << obj; return 0; }

Задача 9.11. Добавьте пользовательскую функцию ввода в класс strtype Задачи 9.9.

Задача 9.12. Создайте класс для хранения целого и его наименьшего делителя. Создайте для этого класса пользовательские функции ввода и вывода.

Задача 9.13. Напишите программу для вывода числа 100 в десятичной, шестнадцатеричной и восьмеричной системе счисления. (Используйте флаги формата класса ios.)

Задача 9.14. Напишите программу для вывода значения 1000.5364 в 20-символьном поле, с выравниванием влево, с двумя позициями после запятой и символом \* в качестве символа заполнения. (Используйте флаги формата класса ios.)

Задача 9.15. Перепишите ваши ответы на Задачи 9.13 и 9.14 так, чтобы использовались манипуляторы ввода/вывода.

Задача 9.16. Покажите, как записать и как отобразить для класса cout флаги формата. Используйте функции-члены либо манипуляторы.

Задача 9.17. Создайте для следующего класса пользовательские функции ввода и вывода: class pwr { int base; int exponent; double result; // base to the exponent power public: pwr(int b, int e); }; pwr::pwr(int b, int e) { base = b; exponent = e; result = 1; for ( ; e; e--) result = result \* base; }

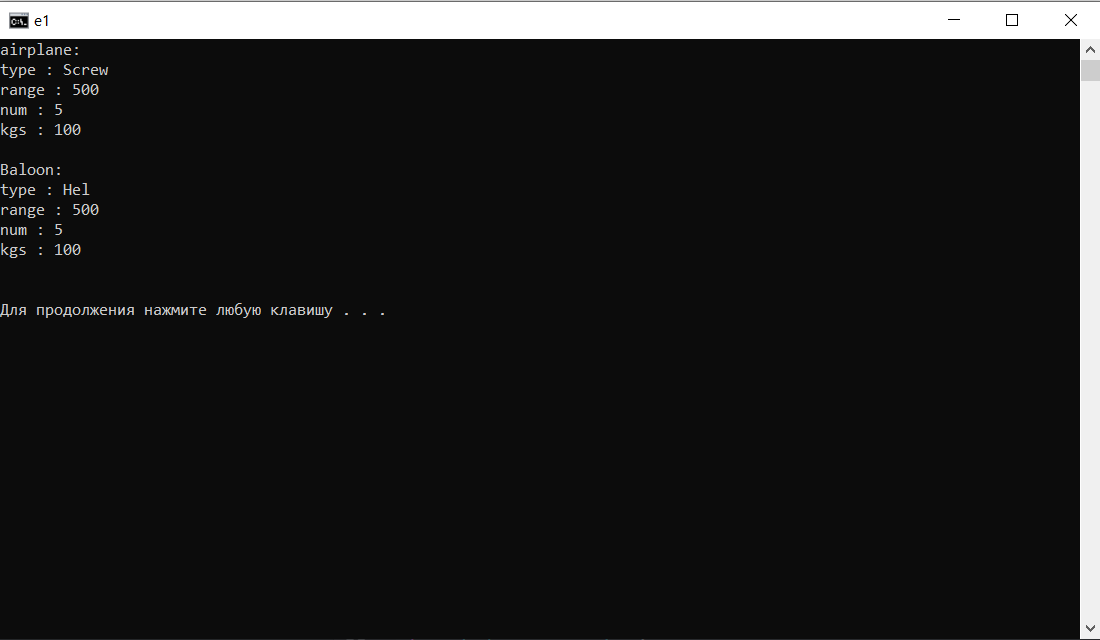
Задача 9.18. Создайте класс box для хранения размеров квадрата. Для вывода изображения квадрата на экран создайте пользовательскую функцию вывода. (Способ изображения выберите любой.)

Задача 9.19. Используя показанный здесь класс stack, создайте пользовательскую функцию вывода для вставки в поток содержимого стека. Покажите, что функция работает. #include using namespace std; #define SIZE 10 class stack { char stck[SIZE]; // holds the stack int tos; // index of top -of - stack public: stack(); void push(char ch); // push character on stack char pop(); // pop character from stack }; // Initialize the stack: stack::stack() { tos = 0; } // Push a character void stack::push (char ch) { if (tos == SIZE) { cout << "Stack is full\n"; return 0; } stck[tos] = ch; tos++; } // Pop a character: char stack::pop() { if (tos==0) { cout << "Stack is empty\n"; return 0; } tos--; return stck[tos]; }

Задача 9.20. Напишите программу с классом watch, который бы играл роль обычных часов. Используя стандартные функции времени, создайте конструктор класса, который должен считывать и запоминать системное время. Для вывода этого времени на экран создайте пользовательскую функцию вывода.

Задача 9.21. На основе класса, созданного для преобразования футов в сантиметры, напишите пользовательскую функцию ввода, формирующую строку-приглашение для записи числа футов. Кроме этого, напишите пользовательскую функцию вывода для отображения на экране как числа футов, так и числа сантиметров. Включите указанные функции в программу и продемонстрируйте их работоспособность. class ft\_to\_cms { double feet; double cms; public: void set(double f) { feet = f; cms = f \* 30.48; } }; Примечание: Фут (обозначается как «ft») — единица измерения длины. Фут равен 0,3048 м и используется в английской системе мер и США.

Завдання 9.1



Код :

#include <iostream>

using namespace std;

enum typeplane { screw , jet };

enum typebaloo { hyd , hel };

class airship{

int num;

int kgs;

public:

airship(int num , int kgs){

this->num = num ;

this->kgs = kgs ;

}

show(){

printf("num : %d\nkgs : %d\n\n",num , kgs );

}

};

class baloon : airship{

char\* type;

int range ;

public:

baloon (int num , int kgs , int type , int range) : airship(num,kgs){

if (type == 0 )this->type = "Hyd" ;

if (type == 1 )this->type = "Hel" ;

this->range = range ;

}

showall(){

printf("Baloon:\ntype : %s\nrange : %d\n",type,range );

show();

}

};

class airplane : airship{

char \* type;

int range;

public:

airplane (int num , int kgs , int type , int range) : airship(num,kgs){

if (type == 0 )this->type = "Screw" ;

if (type == 1 )this->type = "Jet" ;

this->range = range ;

}

showall(){

printf("airplane:\ntype : %s\nrange : %d\n",type,range );

show();

}

};

int main (){

airplane ob1(5,100,screw,500);

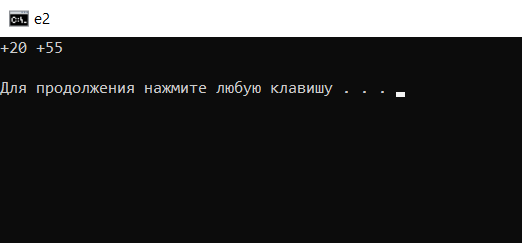
ob1.showall();

baloon ob2(5,100,hel,500);

ob2.showall();

}

Завдання 9.2



Код:

#include <iostream>

using namespace std;

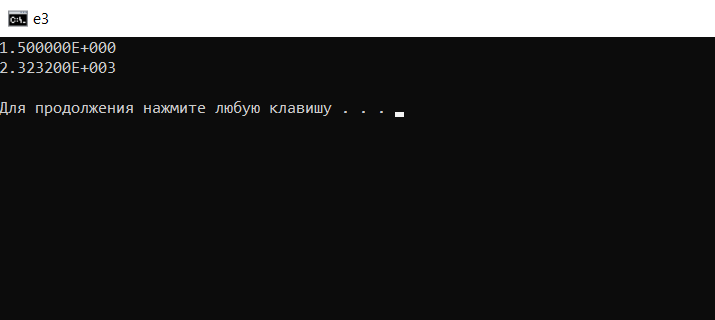
int main (){

cout.setf(ios::showpos );

cout << 20 << " " << 55 <<endl;

}

Завдання 9.3



Код :

#include <iostream>

using namespace std ;

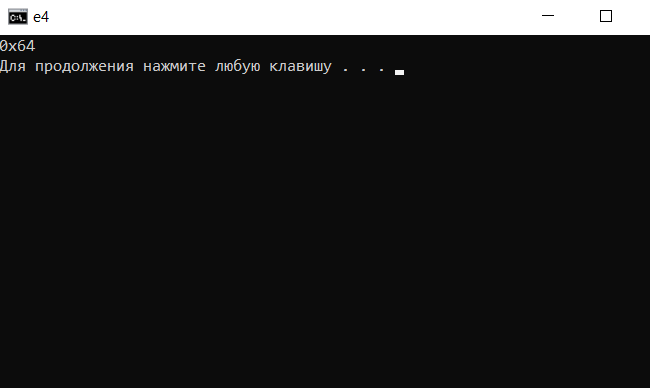
int main(){

cout.setf(ios::showpoint | ios::uppercase | ios::scientific);

cout << 1.5 << endl << 2323.2 << endl;

}

Завдання 9.4



Код :

#include <iostream>

using namespace std;

int main(){

cout.unsetf(ios::dec);

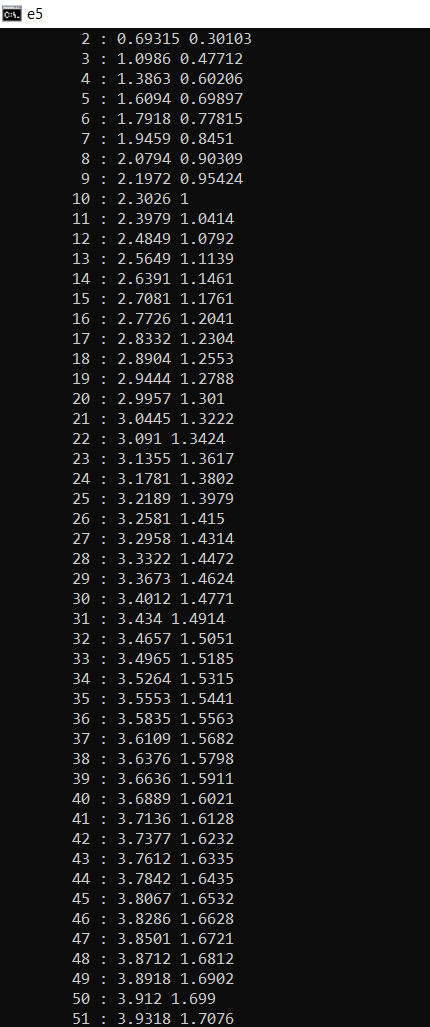
cout.setf(ios::showbase| ios::hex );

cout << 100 ;

cout.unsetf(ios::showbase | ios::hex );

}

Завдання 9.5



Код :

#include <iostream>

#include <cmath>

using namespace std;

int main() {

cout.setf(ios::right);

for (int i = 2; i <= 100; i++) {

cout.width(10);

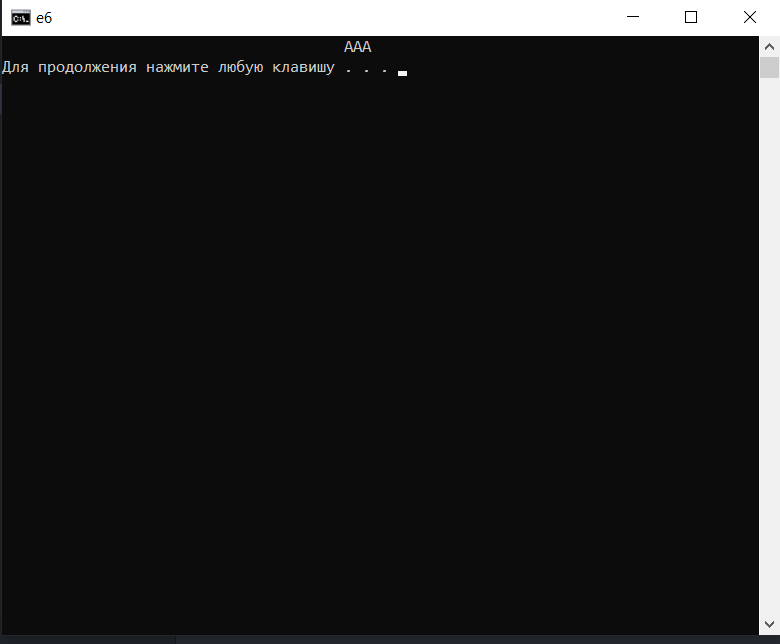
cout.precision(5);

cout << i << " : " << log(i) << " " << log10(i) << endl ;

}

}

Завдання 9.6



Код :

#include <iostream>

#include <cstring>

using namespace std;

center (char \* s){

cout.width(40 + strlen(s)/2);

cout << s ;

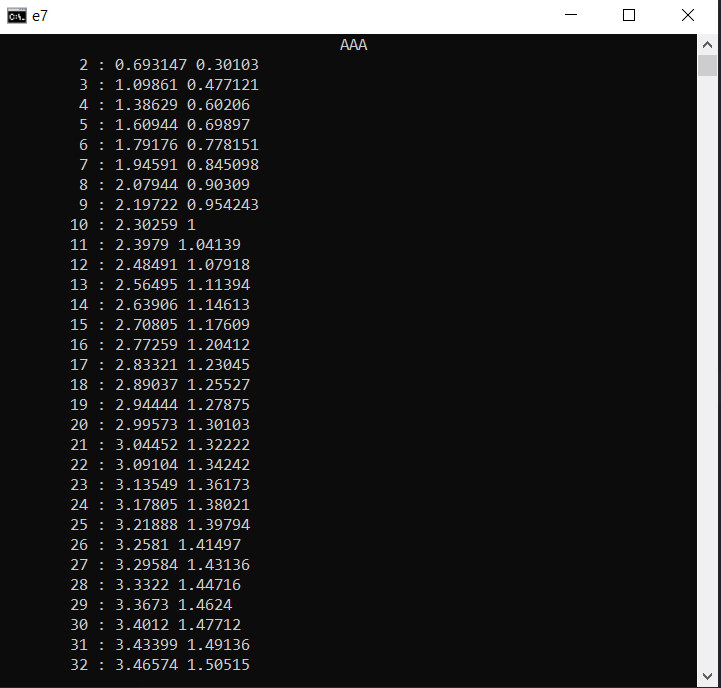
}

int main (){

center ("AAA");

}

Завдання 9.7



Код :

#include <iostream>

#include <cstring>

#include <iomanip>

#include <cmath>

using namespace std;

center (char \* s){

cout << setw(40 + strlen(s)/2) << s << endl ;

}

int main (){

center ("AAA");

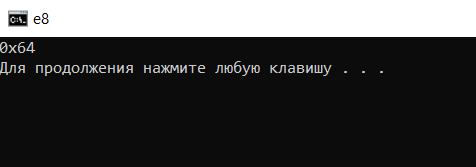
for (int i = 2; i <= 100; i++) {

cout << setw(10) << i << " : " << log(i) << " " << log10(i) << endl ;

}

}

Завдання 9.8



Код :

#include <iostream>

#include <iomanip>

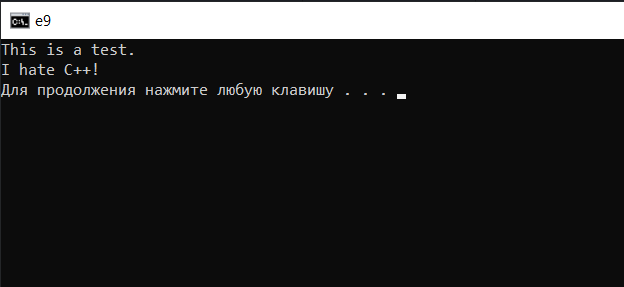
using namespace std;

int main(){

cout << showbase << hex << 100 ;

}

Завдання 9.9



Код :

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

class strtype {

char \*p;

int len;

public:

strtype(char \*ptr);

~strtype() { delete [] p; }

friend ostream &operator <<(ostream &stream, strtype &obj);

};

strtype::strtype(char \*ptr) {

len = strlen(ptr)+1;

p = new char(len);

if (!p) {

cout << "Allocation error\n";

exit(1);

}

strcpy(p, ptr);

}

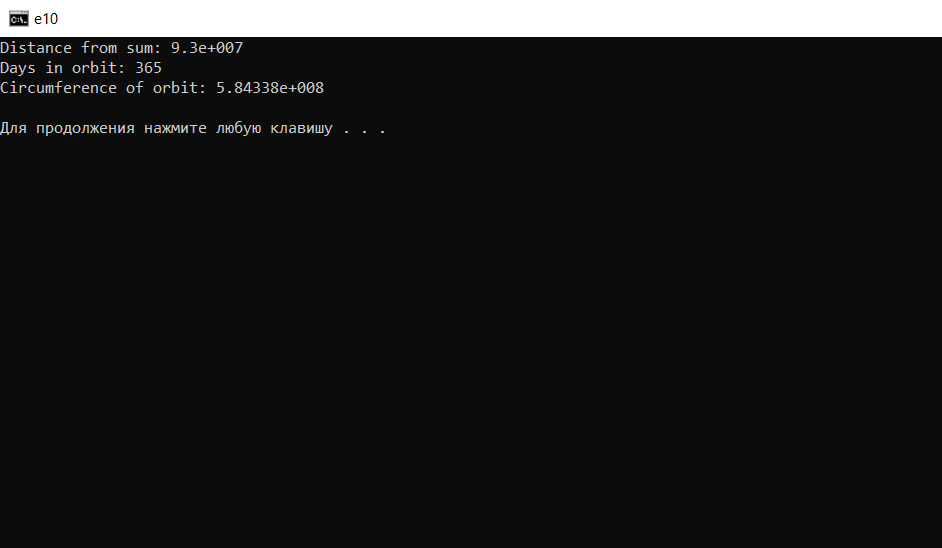
ostream &operator <<(ostream &stream, strtype &obj){

stream << obj.p;

return stream;

}

Завдання 9.10



Код :

#include <iostream>

using namespace std;

class planet {

protected:

double distance; // расстояние в милях от Солнца

int revolve; // полный оборот в днях

public:

planet(double d, int r) {

distance = d;

revolve = r;

}

};

class earth : public planet {

double circumference; // окружность орбиты

public:

earth(double d, int r) : planet(d, r) {

circumference = 2 \* distance \* 3.1416;

}

friend ostream & operator << (ostream & stream , earth &obj){

stream << "Distance from sum: " << obj.distance << '\n';

stream << "Days in orbit: " << obj.revolve << '\n';

stream << "Circumference of orbit: " << obj.circumference << '\n';

return stream;

}

};

int main() {

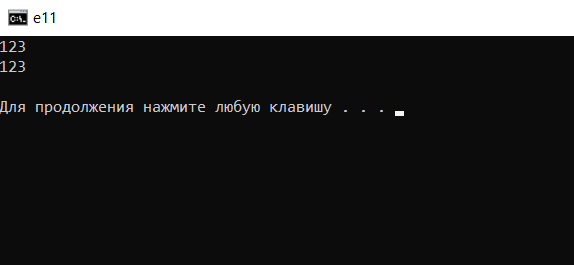
earth obj(93000000, 365);

cout << obj;

return 0;

}

Завдання 9.11



Код :

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

class strtype {

char \*p;

int len;

public:

strtype(char \*ptr);

~strtype() { delete [] p; }

friend ostream &operator <<(ostream &stream, strtype &obj){

stream << obj.p ;

return stream;

}

friend istream &operator >>(istream &stream, strtype &obj){

stream >> obj.p ;

return stream;

}

};

strtype::strtype(char \*ptr = "empty") {

len = strlen(ptr)+1;

p = new char(len);

if (!p) {

cout << "Allocation error\n";

exit(1);

}

strcpy(p, ptr);

}

int main() {

strtype s1;

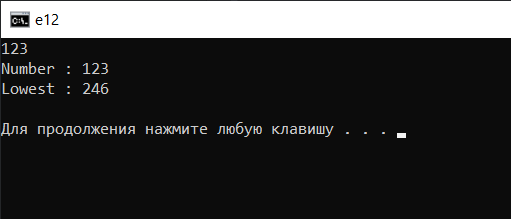
cin >> s1 ;

cout << s1 << '\n';

return 0;

}

Завдання 9.12



Код :

#include <iostream>

using namespace std;

class det\_low{

int num ;

int lowest;

public:

friend istream &operator >>(istream & stream , det\_low &ob){

stream >> ob.num;

int i;

for ( i = ob.num + 1 ; ; i++) {

if(i % ob.num == 0){

ob.lowest = i ;

break;

}

}

return stream;

}

friend ostream &operator <<(ostream & stream , det\_low& obj){

stream << "Number : " << obj.num << endl ;

stream << "Lowest : " << obj.lowest << endl;

return stream;

}

};

int main(){

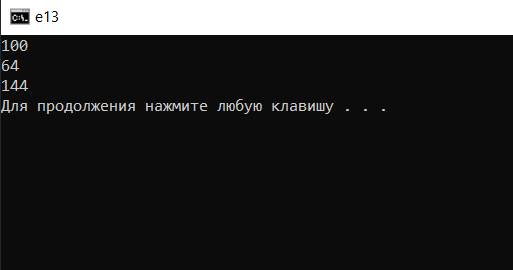
det\_low ob ;

cin >> ob ;

cout << ob ;

}

Завдання 9.13



Код :

#include <iostream>

using namespace std;

int main() {

cout << 100 << endl ;

cout.unsetf(ios::dec);

cout.setf(ios::hex);

cout << 100 << endl ;

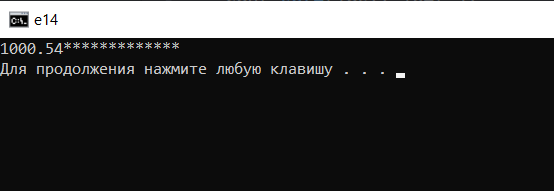
cout.unsetf(ios::hex);

cout.setf(ios::oct);

cout << 100 ;

}

Завдання 9.14



Код :

#include <iostream>

using namespace std;

int main(){

cout.setf(ios:: left );

cout.fill('\*');

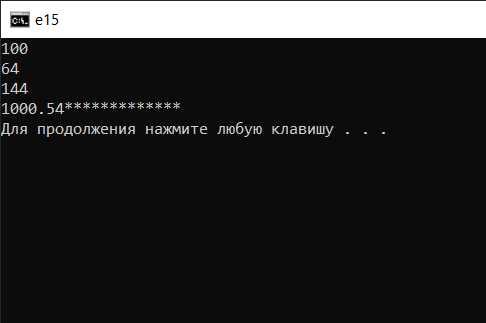
cout.width(20);

cout.precision(6);

cout <<1000.5364 ;

}

Завдання 9.15



Код :

#include <iostream>

#include <iomanip>

using namespace std;

int main(){

cout << 100 << endl ;

cout << hex << 100 << endl ;

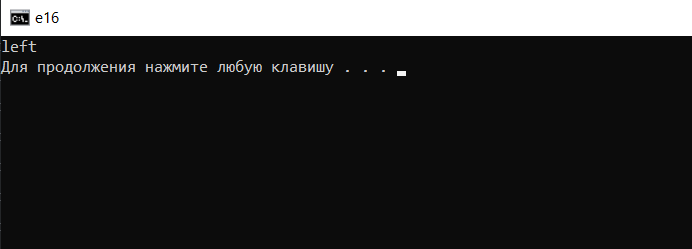
cout << oct << 100 << endl ;

cout.setf(ios:: left );

cout << setw(20) << setprecision(6) << setfill('\*') <<1000.5364 ;

}

Завдання 9.16



Код :

#include <iostream>

using namespace std;

int main(){

ios::fmtflags f ;

cout.setf(ios::left);

f = cout.flags();

if (f & ios::right) {

cout << "right";

}else

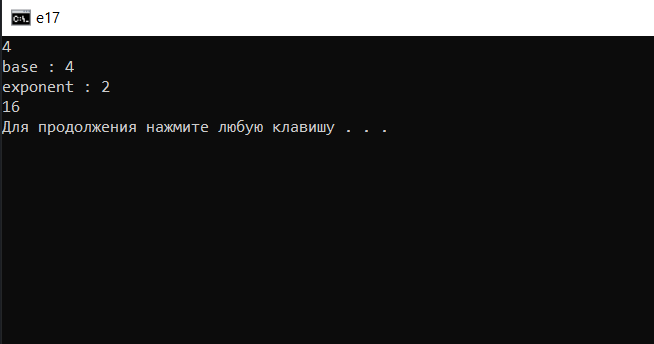
if (f & ios::left) {

cout << "left";

}else cout << "NO";

}

Завдання 9.17



Код :

#include <iostream>

using namespace std;

class pwr {

int base;

int exponent;

double result; // base to the exponent power

public:

pwr(int b, int e);

friend ostream& operator << (ostream & stream , pwr & ob){

stream << ob.result ;

return stream;

}

friend istream& operator >> (istream & stream , pwr & ob){

printf("\nbase : ");

stream >> ob.base ;

printf("exponent : ");

stream >> ob.exponent ;

ob.result = 1;

for ( ; ob.exponent; ob.exponent--)

ob.result = ob.result \* ob.base;

return stream;

}

};

pwr::pwr(int b, int e) {

base = b;

exponent = e;

result = 1;

for ( ; e; e--)

result = result \* base;

}

int main (){

pwr obj(2,2);

cout << obj ;

cin >> obj ;

cout << obj ;

}

Завдання 9.18



Код :

#include <iostream>

#include <iomanip>

using namespace std;

class box {

int a ;

public :

box(int side){

a = side ;

}

friend ostream & operator << (ostream & stream , box &ob ){

stream << ' ' << setfill('\_') << setw(ob.a\*2) << ' ' << endl << setfill(' ');

for (int i = 0; i < ob.a; i++) {

stream << '|' << setw(ob.a\*2) << '|' <<endl;

if(i == ob.a-2) cout << setfill('\_');

}

return stream;

}

};

int main(){

box ob1(2);

cout << ob1 ;

box ob2(5);

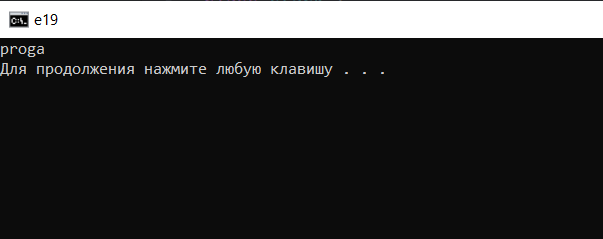
cout << ob2 ;

box ob3(10);

cout << ob3 ;

}

Завдання 9.19



Код :

#include <iostream >

using namespace std;

#define SIZE 10

class stack {

char stck[SIZE];

int tos;

public:

stack();

void push(char ch);

char pop();

friend ostream & operator << (ostream & stream , stack& ob){

for (int i = ob.tos-1; i >= 0; i--) {

stream << ob.stck[i];

}

return stream ;

}

};

stack::stack() {

tos = 0;

}

void stack::push (char ch) {

if (tos == SIZE) {

cout << "Stack is full\n";

exit(0);

}

stck[tos] = ch;

tos++;

}

char stack::pop() {

if (tos==0) {

cout << "Stack is empty\n";

exit(0);

}

tos--;

return stck[tos];

}

int main(){

stack ob;

ob.push('a');

ob.push('g');

ob.push('o');

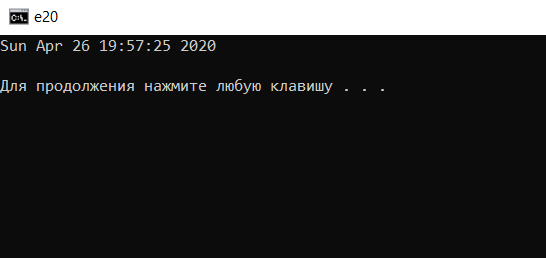
ob.push('r');

ob.push('p');

cout << ob;

}

Завдання 9.20



Код :

#include <iostream>

#include <ctime>

using namespace std ;

class watch {

time\_t t ;

public:

watch(){

t = time(NULL);

}

friend ostream & operator << (ostream& stream , watch & ob){

stream << ctime(&ob.t) ;

return stream ;

}

};

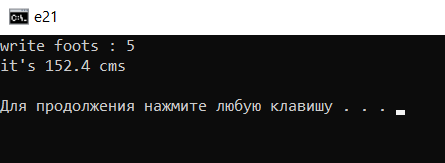
int main (){

watch ob ;

cout << ob ;

}

Завдання 9.21



Код :

#include <iostream>

using namespace std ;

class ft\_to\_cms {

double feet;

double cms;

public:

void set(double f = 1) {

feet = f;

cms = f \* 30.48;

}

friend istream & operator >> (istream & stream , ft\_to\_cms & ob){

printf("write foots : ");

stream >> ob.feet ;

ob.set(ob.feet);

return stream ;

}

friend ostream & operator << (ostream & stream , ft\_to\_cms & ob){

stream <<"it's "<< ob.cms << " cms"<< endl ;

return stream ;

}

};

int main (){

ft\_to\_cms ob ;

cin >> ob ;

cout << ob ;

}