អំពីការសរសេរកម្មវិធីជាភាសា​ C++

១. ទំរង់នៃកម្មវិធីជាភាសា C++

ដើម្បីចាច់ផ្តើមសិក្សាអំពីការសរសេរកម្មវិធី យើងពិនិត្យមើលកម្មវិធីខាងក្រោមនេះ​។

// first program in c++

#include<iostream.h>

Int main()

{

cout<<”Hello World!”;

return 0;

}

ឧបមាថាកម្មវិធីខាងលើនេះគេបានសរសេរដាក់នៅក្មុង file **​**មួយឈ្មោះ hiworld.cpp ហើយក្រោយពី Compile បានជោគជ័យ និង​ ​Run វាគេទទួលបានលទ្ធផល Hello World បង្ហាញ​ លើអរក្រង់​។ ប៉ុន្តែអ្វីដែលសំខាន់នៅក្នុងកម្មវិធីនេះ យើងគូរយល់នូវចំនុចសំខាន់ៗមួយចំនួន​ ដូចខាងក្រោម  ៖

// first program in c++

ឃ្លានេះជា comment ។​ គ្រប់ឃ្លាទាំងអស់ដែលចាប់ផ្តើមដោយសញ្ញា // ត្រូវបានចាត់ទុកជា comment ហើយវាពុំមានឥទ្ធិពលលើកម្មវីធីឡើយ។ វាត្រូវបានគេប្រើសម្រាប់ពន្យល់លើផ្នែកណា​មួយនៃកម្មវិធី​។

#include <iostream.h>

ឃ្លាដែលចាប់ផ្តើមដោយសញ្ញា # គឺជា directives របស់ preprocessor ។ វាប្រាប់ទៅ compiler ដើម្បីបញ្ចូល header file ទូទៅឈ្មោះ iostream នេះ ពីព្រោះមុខងាររបស់វាទាក់ទងទៅ​ នឹងការបញ្ចូលនិងបង្ហាញតម្លៃរបស់ c++ ហើយនិងមានការប្រើនៅលើកក្រោយទៀត។

int main()

ឃ្លានេះឆ្លើយតបទៅនឹងការចាប់ផ្តើមនៃការប្រកាសរបស់ main() function ។ វាជាចំណុច​ ដែលកម្មវិធីភាសា C++ ចាប់ផ្តើមធ្វើការប្រតិបត្តិ។ ទោះជា main() function សរសេរនៅខាងលើ គេ កណ្តាល ខាងក្រោមនៃ code ក៏ដោយ។ codes ដែលនៅក្នុង main() ជានិច្ចជាកាលប្រតិបត្តិ មុនគេ ពេលកម្មវិធីបានចាប់ផ្តើមដំណើរការ។

នៅជាប់ខាងក្រោយ main()​ មានសញ្ញាវង់ក្រចក ព្រោះវាជា​​ function ។ គ្រប់ functions ទាំងអស់របស់ភាសា C++ មានសញ្ញាវងក្រចក និងអាចមាន arguments គ្មាននៅក្នុងវង់ក្រចក នោះ។ ហើយនៅជាប់នឹងវង់ក្រចកនេះវាមានសញ្ញា { } ដើម្បីអមនូវ codes ទាំងឡាយណាដែល ជារបស់ main() function នោះ។

cout << “Hello World”;

ឃ្លានេះជាឃ្លាសំខាន់នៅក្នុងកម្មវិធីនេះ។​ cout ជា​ stream សម្រាប់អោយ c++ បញ្ចេញ លទ្ធផលមកលើអេក្រង់ នៅក្នុងករណីនេះវាបង្ហាញអក្សរមួយឃ្លាគឺ Hello World មកលើអេក្រង់ cout ត្រូវបានគេប្រកាស header file មួយគឺ iostream.h ហេតុនេះហើយដើម្បីប្រើវាបានគេត្រូវ បញ្ចូលឃ្លា #include<iostram.h>។

គួរកត់សម្គាល់ថា នៅខាងចុងនៃឃ្លានេះបានបញ្ចប់ដោយសញ្ញា ; ។ សញ្ញានេះសម្គាល់ ការបញ្ចប់នៃឃ្លាទាំងឡាយរបស់ភាសា c++ ។

return 0;

ពាក្យ return នាំអោយ main() function ទៅរកការបញ្ចប់ ហើយត្រឡប់ទៅកាន់ code ដែល ពាក្យនេះបានបញ្ជាក់ប្រាប់ ក្នុងករណីនេះគឺជាលេខ​សូន្យ។ នេះជាមធ្យោបាយមួយសម្រាប់ បញ្ចប់កម្មវិធីដែលពុំមាន error កើតឡើងនៅពេលប្រតិបត្តិការ។

កម្មវិធីត្រូវបានរៀបជាទម្រង់នៅលើបន្ទាត់ផ្សេងគ្នា ដើម្បីងាយស្រួលអាន ប៉ុន្តែវាមិនមែនជាករណីចាំបាច់ឡើយ។

ឧទាហរណ៏៖

#include<iostream.h>

int main() {

cout <<”Hello World”;

return 0;

}

គេអាចសរសេរជាទម្រង់៖

int main() { cout <<”Hello World”; return 0; }

នៅលើបន្ទាត់តែមួយ ហើយវាមានលទ្ធផលដូចគ្នាបេះបិទទៅនឹងកម្មវិធីខាងលើដែរ។

នៅក្នុងភាសា c++ ការខ័ណ្ឌពីឃ្លាមួយទៅឃ្លាមួយទៀតដោយប្រើសញ្ញា​ ; នៅខាងចុង ឃ្លា។ ការសរសេរខ័ណ្ឌ codes អោយនៅជាជួរផ្សេងគ្នា គឺគ្រាន់តែមានប្រយោជន៏ដល់មនុស្ស​ ងាយយល់ និងងាយអានវាប៉ុណ្ណោះ។

// second program in c++

void main()

{

cout<< “Hello World”;

cout<<”It’s a c++ program”;

}

២. លក្ខណ:របស់​ iostream.h

ភាសា c++ ផ្តល់នូងមធ្យោបាយថ្មីមួយសម្រាប់ធ្វើការបញ្ចូលទិន្នន័យនិងបង្ហាញព័ត៌មាន តាមរយ: iostream ។ iostream.h ជា​ header files សម្រាប់បញ្ចេញបញ្ចូល stream ដែលក្នុងនោះ មាន function ទូទៅមួយចំនួនត្រូវបានប្រើសំរាប់បញ្ចូលទិន្នន័យ។ I/O stream​​​ គឺជាសេរីនៃតួអក្សរដែលត្រូវបានគេសរសេរសម្រាប់បង្ហាញនៅលើអេក្រង់​ ឬ អានតម្លៃបញ្ចូលតាមរយ: keyboard ។ ការប្រតិបត្តិបញ្ចូលទិន្នន័យ និងបង្ហាញព័ត៌មានជាទូទៅនៅក្នុង c++ គឺធ្វើឡើងដោយប្រើ cin និង​ cout ។

ភាសា c++ ផ្តល់នូវការបញ្ចូលទិន្នន័យ និងបង្ហាញព័ត៌មានទាក់ទងទៅនឹងលេខ តួអក្សរ និង string ដែលបង្កភាពងាយស្រួល និងមានប្រសិទ្ធភាពខ្ពស់ជាងភាសា c ។

. ios.h : វាត្រូវគ្នានឹង stream សម្រាប់បញ្ចូលទិន្នន័យ និងបង្ហាញព័ត៌មាន។

. istream : វាមាន ​functions សម្រាប់អានបញ្ចូល stream ជាតួអក្សរពី keyboard ។

. ostream : វាមាន​ function សម្រាប់សរសេរបញ្ចេញតួអក្សរទៅលើអេក្រង់។

. iostream : វាមាន​ function នៃ stream សម្រាប់អានបញ្ចូល stream ជាតួអក្សរពី keyboard និងបង្ហាញ stream របស់ object ទៅលើអេក្រង់។

តាមពិតភាសា c++ ពុំមាន function ស្រាប់សម្រាប់បញ្ចូលទិន្នន័យនិងបង្ហាញព័ត៌មានទេ ប៉ុន្តែវាផ្តល់ stream ជាច្រើនប្រភេទផ្សេងៗគ្នាដែលមានទាក់ទងទៅនឹង header files ជាហេតុបង្ក​អោយមានលក្ខណ:ងាយស្រួល។ library របស់ stream គឺជាចំណាត់ថ្នាក់នៃ classes ទាំងឡាយ។

៣.​ ការទំនាក់ទំនងគ្នាតាមរយ: console

Console គឺជា interface មូលដ្ឋានរបស់កុំព្យូទ័រតាមធម្មតាគឺសំដៅទៅលើ​ keyboard និង អេក្រង់។ keyboard ជាឧបករណ៏សំរាប់បញ្ចូលទិន្នន័យ ហើយអេក្រង់ជាឧបករណ៏សម្រាប់ បង្ហាញព័ត៌មាន។

នៅក្នុង iostream c++ library អាចអោយគេប្រើបាន​ data stream ពីរប្រភេទ​ គឺ cin និង​ cout សម្រាប់ធ្វើការប្រតិបត្តិបញ្ចូលទិន្នន័យនិងបង្ហាញព័ត៌មានអោយកម្មវិធី។

៣.១ ការបង្ហាញព័ត៌មានមកក្រៅ (cout)

cout ត្រូវបានគេប្រើសម្រាប់បង្ហាញ object មួយនៅលើឧបករណ៏មួយដែលជាទូទៅគឺ សំដៅទៅលើអេក្រង់។ សញ្ញា << ត្រូវបានគេប្រើជាមួយនឹង cout​​ ។ សញ្ញា << នេះហៅថា insertion operator ។

ទំរង់ទូទៅរបស់​ cout គឺ៖​

cout<<variable1<<variable2<< …<<variable n;

ឧទាហរណ៏

#include<iostream.h>

void main()

{

int x=12;

float y=34.50;

char ch=’a’;

char name[12]=”Ly sokun”;

cout<<”x =”<< x<<’\n’;

cout<<”Y=”<< y<<’\n’;

cout<<”ch=”<< ch<<’\n’;

cout<<”Name =”<< name<<’\n’;

}

៣.២ ការបញ្ចូលទិន្នន័យ (cin)

cin ត្រូវបានប្រើប្រាស់សម្រាប់អានលេខ តួអក្សរ​ ឬអក្សរមួយជួរពីឧបករណ៏បញ្ចូល ទិន្នន័យ តាមធម្មតាវាជា keyboard ។​ សញ្ញា >> ត្រូវបានប្រើប្រាស់ជាមួយនឹង​ cin ។​ សញ្ញា​ >> នេះហៅថា extraction operator ។​

ទំរង់ទូទៅរបស់ cin គឺ

cin>> variable1 >> variable2 >> …variable n;

ឧទាហរណ៏

#include<iostream.h>

void main()

{

int id;

char sex;

float score;

cout<<”input id=”; cin>>id;

cout<<”input sex=”; cin>>sex;

cout<<”input score=”; cin>>score;

cout<<id<<’\t’<<sex<<’\t’<<name;

}

៣.២.១ ការប្រើ cin.get()

ជាមួយនឹង get() function គេប្រើសម្រាប់ទទួលយកតម្លៃជាតួអក្សរ ហើយអាចប្រើបាន ពីរទំរង់។ get() ដែលប្រើដោយគ្មានប៉ារ៉ាម៉ែត្រ នៅក្នុងករណីនេះគេប្រើតម្លៃដែលទទួលបានឬ អាចប្រើបានតាមរយ:​ reference ចំពោះតួអក្សរនោះ។ វាអោយតម្លៃជាតួអក្សរដែលវាប្រទះឃើញឬអោយតម្លៃជា EOF (end of file) បើសិនេជាវាដល់ទីបញ្ចប់នៃ file ។ get() ដែលគ្មាន ​​ ប៉ារ៉ាម៉ែត្រមិនសូវមានការប្រើច្រើនទេ ។​ វាពុំអាចភ្ជាប់ការបញ្ចូលទិន្នន័យតៗគ្នាច្រើនដោយ​សារការអោយតម្លៃរបស់វាមិនមែនជា​ object នៃ iostream ។ ហេតុនេះ ឃ្លាខាងក្រោមនេះមិន ដំណើរការឡើយ។

cin.get() >> myvarone>>myvartwo; // illegal

ការអោយតម្លៃរបស់​ ​cin.get()​ >>myvarone គឺជាចំនួនគត់មិនមែនជា object នៃ iostream នោះទេ។

// Using get() with no parameters

#include<iostream.h>

void main()

{

char ch;

while((ch=cin.get())!=EOF)

{

cout <<” ch: “<< ch <<endl;

}

cout<<” \nDone!\n”;

}

ការប្រើ get() ដោយមានប៉ារ៉ាម៉ែត្រមួយជាតួអក្សរសម្រាប់បញ្ចូលចូលតាម get() function ទៅក្នុង input stream បន្តបន្ទាប់គ្នា។ តម្លៃទទួលបានគឺជា​ object នៃ​ iostream ។ ដូចនេះទម្រង់នេះអាចប្រើភ្ជាប់តៗគ្នាបានដូចបង្ហាញក្នុងកម្មវិធីខាងក្រោម

// Using get() with parameters

#include<iostream.h>

void main()

{

char a, b, c;

cout<<”Enter three letters:”;

cin.get(a).get(b).get(c);

cout<<”a : “<<a<<”\nb: “<<b<<”\nc: “<<c<<endl;

}

៣.៣.២ ការប្រើ cin.get() ជាមួយនឹងការប្រើ cin.seekg()

seekg() ជា function មួយរបស់ input stream(cin) និងជា pointer មួយមានឈ្មោះថា get pointer ហើយត្រូវបានប្រើសម្រាប់បញ្ជាក់ទីតាំងប្រតិបត្តិការបញ្ចូលទិន្នន័យដែលនៅបន្ទាប់ក្នុង stream នោះ។ seekg() រំកិល get pointer (intput) ទៅកាន់ទីតាំងកំណត់។

វាមានទំរង់ដូចខាងក្រោម

Seekg(offset, refposition);

ប៉ារ៉ាម៉ែត្រ​ offset តាងអោយចំនួន byte ដែល pointer ត្រូវផ្លាស់ទីពីទីតាំងកំណត់ដោយ ប៉ារ៉ាម៉ែត្រ refposition ។ refpostion មានតម្លៃថេរកំណត់ដោយ​ ​ios class ជា

ios::beg(=0) ចំណុចចាប់ផ្តើមនៃ stream ឬ file

ios::cur(=1) ទីតាំងកំពុងប្រើរបស់ pointer

ios::end(=2) ចំណុចខាងចុងបំផុតនៃ stream ឬ file

#include<iostream.h>

void main()

{

int id, age;

char name[30], sex[10], add[40];

cout<<”Enter id :”;

cin>>id;

cout<<”Enter age:”;

cin>>age;

cout<<”Enter name: “;

cin.get(name,30); cin.seekg(0);

cout<<”Enter sex: “;

cin.get(sex,10); cin.seekg(0);

cout<<”Enter address:”;

cin.get(add,40); cin.seekg(0);

cout<< id<< “ “<<name<<” “ <<sex<<” “<<age<<” “<<add<<endl;

}

៣.២.៣ ការអានបញ្ចូល string ដោយប្រើ cin.getline()

getline() function ប្រតិបត្តិស្រដៀងទៅនឹង get() function ដែរហើយវាបញ្ចូលនូវតួអក្សរ null នៅពីខាងក្រោយជួរបន្ទាត់ក្នុង array ដែលផ្ទុកតួអក្សរ។ getline() function លប់បំបាត់ចោលនូវតួអក្សរបញ្ចប់ចេញពី stream (បានន័យថា អានតួអក្សរហ់យបោះបង់ចោល) ប៉ុន្តែវាពុំបានផ្ទុក តួអក្សរបញ្ចប់នៅក្នុង array ឡើយ។ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការប្រើ getline() function ។

#include<iostream.h>

void main()

{

const int size = 80;

char buffer[size];

cout<<”Enter a sentence:”<<”\n”;

cin.getline(buffer,size);

cout<<”\n The sentence entered is: “<<”\n”<< buffer<<”\n”;

}

៣.៣ ការប្រើ cerr

ក្រៅពី cout ភាសា​ c++ ផ្តល់នូវ object ផ្សេងទៀតនៃ class ostream ហៅថា cerr ។​ គេប្រើ វាសម្រាប់បង្ហាញព័ត៌មាន error ទៅកាន់ឧបករណ៏ផ្សេងៗទៀត។ ការប្រើ object នេះត្រូវបានបង្ហាញដូចតទៅ

#include<iostream.h>

void main()

{

cout<<”My name is Nasun! Using cout \n”;

cerr<<”many greetings to you! Using cerr \n”;

cerr<<”Hello , Nasun! Using cerr\n”;

}

៤. ការប្រើ Manipulator functions

Manipulator function គឺជា stream function ពិសេសដែលធ្វើអោយផ្លាស់ប្តូរលក្ខណ:នៃ ការបញ្ចូល និងការបង្ហាញទិន្នន័យ។​ វាធ្វើអោយផ្លាស់ប្តូរសញ្ញាទម្រង់ (format flat) និងតម្លៃ​ សម្រាប់ stream ។ ផលប្រយោជន៏ចំបងនៃការប្រើ manipulator functions គឺជាជួយសំរួលនូវ​ ​ ទម្រង់បញ្ចូល streams និងបញ្ចេញ streams ។ ដើម្បីធ្វើការប្រតិបត្តិជាមួយនឹង manipulation functions ទាំងនេះក្នុងកម្មវិធី c++ គេត្រូវតែបញ្ចូល <iomanip.h> header file ។

៥. ការប្រើ manipulators ដែលមានស្រាប់

៥.១ ការប្រើ endl

Endl គឺប្រើសម្រាប់ឲ្យចុះដើមបន្ទាត់។ គេប្រើវាញឹកញាប់ណាស់ក្នុងភាសា c++ ។

ឧទាហរណ៏ទី១ cout << “ a “ << endl << “b “<< endl ;

ឧទាហរណ៏ទី២ cout << “ a= “ << a << endl;

cout << “ b = “ << b <<endl;

#include<iostream.h>

void main() {

cout << “My name is Nasun”;

cout << endl;

cout << “Many greeting to you”;

}

#include<iostream.h>

void main() {

int a;

cout <<” a = “<< a << endl;

cout <<” a =”<< a << ‘\n’;

cout << “a = “<< a <<”\n”;

}

៥.២ ការប្រើ setbase()

Setbase() manipulator ប្រើសម្រាប់បំលែងនូវតម្លៃលេខពីប្រព័ន្ធលេខណាមួយទៅជា ប្រព័ន្ធលេខណាមួយផ្សេងទៀត។ គេមាន manipulators មួយចំនួនទៀតដែលប្រើសម្រាប់​ ​​ បំលែងលេខតាមប្រព័ន្ធទូទៅនៅក្នុងភាសា c++ ។

dec - លេខប្រព័ន្ធគោលដប់

hex - លេខប្រព័ន្ធគោលដប់ប្រាំមួយ

oct - លេខប្រព័ន្ធគោលប្រាំបី

លំនាំនៃការប្រើ setbase() ត្រូវបានបញ្ចូលនូវ header file ឈ្មោះ iomanip.h ។ ការបញ្ចូល តម្លៃ ឬការបង្ហាញលទ្ធផលតាមធម្មតាគឺជាលេខប្រព័ន្ធគោលដប់ (dec)។

ឧទាហរណ៏

//Using dec, hex , oct manipulator

#include<iostream.h>

#include<iomanip.h>

void main() {

int value;

cout << “Enter value : “;

cin >> value;

cout << “Decimal base = “<< dec << value << endl;

cout << “Hexadecimal base = “<< hex << value <<endl;

cout << “Octal base =”<< oct << value << endl;

}

//Using setbase manipulatior

#include<iostream.h>

#include<iomanip.h>

void main() {

int value;

cout << “Enter value : “;

cin >> value;

cout << “Decimal base = “<< setbase(10);

cout << value <<endl;

cout << “Hexadecimal base =”<< setbase(16);

cout << value<<endl;

cout<< “Octal base =”<< setbase(8)<<value<<endl;

}

៥.៣ ការប្រើ setw()

setw() មកពីពាក្យ set width ។ setw() manipulator ត្រូវបានគេប្រើសម្រាប់កំណត់ ចំនួន​ អប្បបរមានៃទីតាំងតូអក្សរដែលអញ្ញាតមួយត្រូវការបង្ហាញមកក្រៅ។ ទម្រង់ទូទៅរបស់វាគឺ

setw(int w)

ដេល w ជាទំហំមួយមានតម្លៃ default ស្មើនឹងសូន្យ ហើយវាត្រូវការកំណត់ទំហំសម្រាប់តែការ បញ្ចូលតម្លៃដែលនៅបន្ទាប់ប៉ុណ្ណោះ។

//Using setw manipulator

#include<iostream.h>

#include<iomanip.h>

void main() {

int a ,b;

a = 200;

b = 300;

cout << setw(5) << a << setw(5) << b << endl;

cout << setw(6) << a << setw(6) << b << endl;

cout << setw(7) << a << setw(7) << b << endl;

cout << setw(8) << a << setw(8) << b << endl;

}

៥.៤ ការប្រើ setfill()

setfill() ត្រូវបានគេប្រើសម្រាប់កំណត់តួអក្សរផ្សេងគ្នា ដើម្បីបំពេញនូវទំហំទីតាំងនៃ តម្លៃរបស់អញ្ញាតដែលមិនបានប្រើ។

/Using setfill manipulator

#include<iostream.h>

#include<iomanip.h>

void main() {

int a ,b;

a = 200;

b = 300;

cout << setfill(‘ \* ‘);

cout << setw(5) << a << setw(5) << b << endl;

cout << setw(6) << a << setw(6) << b << endl;

cout << setw(7) << a << setw(7) << b << endl;

cout << setw(8) << a << setw(8) << b << endl;

}

៥.៥ ការប្រើ setprecision()

setprecision() ប្រើសម្រាប់កំណត់ចំនួនខ្ទង់លេខនៃការបង្ហាញលទ្ធផលរបស់លេខទសភាគមួយ។ setprecision() ត្រូវបានបង្កើតឡើងនៅក្នុង <iomanip.h> header file ។ ទម្រង់ទូទៅគឺ

setprecistion ( int p)

ដែលកំណត់ចំនួនខ្ទង់លេខទសភាគតាមរយ:​ p ។ តម្លៃ default របស់វាគឺ៦។

//Using setprecision manipulator

#include<iostream.h>

#include<iomanip.h>

void main() {

int a ,b,c ;

a = 5;

b = 3;

c = a/b;

cout << setprecision(1) << c << endl;

cout << setprecision(2) << c << endl;

cout << setprecision(3) << c << endl;

cout << setprecision(4) << c << endl;

cout << setprecision(5) << c << endl;

cout << setprecision(6) << c << endl;

}

//Using setprecision manipulator

#include<iostream.h>

#include<iomanip.h>

void main() {

float num = 123.45678;

cout << num << endl;

cout << setprecision(7) << num << endl;

cout << setprecision(2) << num << endl;

}

៥.៦ ការប្រើ ends

ends គឺជា manipulator មួយប្រើសម្រាប់ភ្ជាប់តួអក្សរ ‘\0’ នៅខាងចុងនៃ string ។​

​​​​//Using ends manipulator

#include<iostream.h>

void main() {

int num=1234;

cout << ‘ \” ‘ << “ number = “ << number << ends;

cout << ‘\t “ ‘ << endl;

}

៥.៧ ការប្រើ setiosflags និង​ resetiosflags

Setiosflags manipulator function ប្រើសម្រាប់ផ្ទៀងផ្ទាត់ការកំណត់លទ្ធផល បញ្ចេញមក ក្រៅ និងការបញ្ចូលទិន្នន័យផ្សេងគ្នា។ វាធ្វើសកម្មភាពដូចទៅនឹង setf() function ដែរ​។

ទម្រង់ទូទៅរបស់វាគឺ

setiosflags(long f)

//Using setiosflags of basefield

#include<iostream.h>

#include<iomanip.h>

void main() {

int value;

cout << “Enter a number : “;

cin >> value ;

cout << setiosflags(ios::showbase);

cout << setiosflags(ios::dec);

cout << “Decimal =” << value << endl;

cout << setiosflags(ios::hex);

cout << “Hexadecimal = “ << value << endl;

cout << setiosflags(ios::oct);

cout << “Octal = “ << value <<endl;

}

៦. ការបញ្ចូល​ និង បង្ហាញលទ្ធផល stream flags

ដើម្បីប្រើ manipulators ជាច្រើននៅខាងលើនេះ អញ្ញាតដែលជាសញ្ញាសម្គាល់ (flags field) បញ្ជាក់អោយការកំណត់យកលក្ខណ:ណាមួយ។ តារាងខាងក្រោមនេះបង្ហាញឈ្មោះសម្គាល់ (flags names) និងអត្ថន័យរបស់វា។

|  |  |
| --- | --- |
| ឈ្មោះសម្គាល់ | អត្ថន័យ |
| right | បង្ហាញលទ្ធផលដោយតំរឹមខាងស្តាំ |
| left | បង្ហាញលទ្ធផលដោយតំរឹមខាងឆ្វេង |
| internal | ដាក់សញ្ញាប្រភេទណាមួយពីក្រោយសញ្ញា +/- ឬសញ្ញាសម្គាល់ប្រព័ន្ធ គោលណាមួយ។ |
| dec | លេខគោលដប់ (បង្ហាញចំនួនគត់ក្នុងទំរង់លេខគោលដប់) |
| oct | លេខគោលប្រាំបី (បង្ហាញចំនួនគត់ក្នុងទំរង់លេខគោលប្រាំបី) |
| hex | លេខគោលដប់ប្រាំមួយ(បង្ហាញចំនួនគត់ក្នុងទំរង់លេខគោលដប់ប្រាំមួយ) |
| showbase | បង្ហាញលេខគោលប្រាំបី និង លេខគោលដប់ប្រាំមួយ។ |
| showpoint | បង្ហាញលេខជាទសភាគ |
| uppercase | បង្ហាញជាលក្ខណ:អក្សរធំសម្រាប់លេខគោលដប់ប្រាំមួយ |
| showpos | បង្ហាញ ‘+’ ចំពោះចំនួនគត់វិជ្ជមាន |
| scientific | ប្រើ E សម្រាប់សម្គាល់លេខទសភាគ |
| fixed | ប្រើសញ្ញាសម្គាល់លេខទសភាគ |

៦.១ ការកំណត់យកលក្ខណ: bit format flag

ដើម្បីប្តូរសភាពនៃ​ cout object នោះ​ bits ដែលតាងអោយសភាពរបស់វាត្រូវតែធ្វើការ ផ្លាស់ប្តូរ។ setf() function សម្រាប់កំណត់​ bit format flags របស់​ i/o stream ។

ទម្រង់ទូទៅនៃ setf() function គឺ

​Cout.setf(flags ដែលត្រូវធ្វើការកំណត់យកលក្ខណ:ណាមួយ)

ឧទាហរណ៏ cout.setf(ios::showbase);

គេអាចប្រើឈ្នាប់ឬ (|) នៅក្នុង argument នៃ setf() function ដើម្បីធ្វើការផ្លាស់ប្តូរ bit format flag អោយមានសភាពលក្ខណ:ច្រើន។

ឧទាហរណ៏

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

៦.២ ការបំបាត់លក្ខណ: bit format flag