ការប្រើឃ្លាប្រតិបត្តិតាមល័ក្ខខ័ណ្ឌ​

១. ទម្រង់ប្រតិបត្តិទៅតាមល័ក្ខខ័ណ្ឌ

កន្សោមល័ក្ខខ័ណ្ឌត្រូវបានប្រើប្រាស់សម្រាប់ធ្វើការសម្រេចចិត្តក្នុងការប្រតិបត្តិទម្រង់ ផ្សេងៗនៃឃ្លាប្រតិបត្តិតាមល័ក្ខខ័ណ្ឌ និងកន្សោមរបស់វានឹងធ្វើការពន្យល់មានដូចខាងក្រោម

១.១ ការប្រើឃ្លា if

ឃ្លា if ត្រូវបានប្រើសម្រាប់បញ្ញាក់នូវកន្សោមល័ក្ខខ័ណ្ឌ។ ប្រសិនបើកន្សោមល័ក្ខខ័ណ្ឌ មានតម្លៃពិតនោះ វានឹងប្រតិបត្តិការនូវឃ្លាផ្សេងៗដែលស្ថិតក្រោមល័ក្ខខ័ណ្ឌ ផ្ទុយទៅវិញវានឹង ប្រតិបត្តិការនូវឃ្លាដែលមានលក្ខណ:ជម្រើស (optional) ។​ សញ្ញា {​ និង }​ ត្រូវបានប្រើសម្រាប់ បណ្តុំឃ្លា និងការប្រើប្រាស់ផ្សេងៗអោយទៅជាឃ្លាសាំញាំមួយ ឬ block មួយ​។ ឃ្លាសាំញាំ ឬ block ទាំងនេះជានិច្ចជាកាលត្រូវបានចាត់ទុកដូចជាឃ្លាតែមួយគត់។ ទម្រង់ងាយបំផុតរបស់វា មានដូចតទៅ

if(condition) {

statement1 ;

statement2 ;

…….

…….

}

ឧទាហរណ៏ទី១

int a = 40;

int b = 30;

if ( a > b)

cout << “Largest value =”<< a;

ឧទាហរណ៏ទី២

#include<iostream.h>

void main() {

float account\_balance;

cout <<”What is your account balance?”;

cin >> account\_balance;

if (account\_balance < 0)

cout << “Your account is in the red “<< endl;

if (account\_balance >= 0)

cout <<” Your balance is in the black”<<endl;

}

ឧទាហរណ៏ទី៣

#include<iostream.h>

void main() {

float x , y , z, max ;

cout << “Enter x=”; cin >> x;

cout << “Enter y=”; cin >> y;

cout <<”Enter z=”; cin >>z;

max=x;

if (max <y)

max=y;

if​ (max < z )

max=z;

cout << “Maximum is << max<<endl;

}

១.២ ការប្រើ if else

if ​​​(condition)

statement1

else

statement2

ឧទាហរណ៏ទី៤

#include<iostream.h>

void main() {

float x , y ;

cout << “Enter x=”; cin >> x;

cout << “Enter y=”; cin >> y;

if (x > y)

cout <<”Largest value is “<< x << endl;

else

cout <<”Largest value is “<< y << endl;

}

ឧទាហរណ៏ទី៥

#include<iostream.h>

void main() {

float average;

cout << “Enter average =”; cin >> average;

if (average < 50)

cout <<”You are Failed”<<endl;

else

cout <<”You are Passed”<<endl;

}

១.៣ ការប្រើ Nested if else

ទម្រង់ទី១

if (condition1)

if (condition2)

…….

…….

If (condition n)

statement1;

else

statement2;

else

…….

…….

else

statement n;

ឧទាហរណ៏ទី៦

#include<iostream.h>

void main() {

float av;

char grade;

cout <<”Enter average =”; cin >> average;

if(average >=50)

if(average >=65)

if(average >=75)

if(average >=85)

if(average >=95)

grade=’A’;

else

grade= ‘B’;

else

grade=’C’;

else

grade=’D’;

else

grade=’E’;

else

grade=’F’;

cout <<”Grade is =”<< grade<<endl;

}

ទម្រង់ទី២

If(condition1)

statement1;

else if(condition2)

statement2;

……

……

else

Statement n;

ឧទាហរណ៏ទី៧

#include<iostream.h>

void main() {

float av;

char g;

cout <<”Enter average =”; cin >> av;

if(av<50)

g=’F’;

else if(av<65)

g=’E’;

else if(av<75)

g=’D’;

else if(av<85)

g=’C’;

else if(av<95)

g=’B’;

else

g=’A’;

cout << “Grade is =”<< g<<endl;

}

២ ការប្រើឃ្លា switch

ឃ្លា switch គឺជាឃ្លាពិសេសដែលអាចអោយគេប្រតិបត្តិតាមតម្លៃណាមួយនៃ ល័ក្ខខ័ណ្ឌ ដែលត្រូវគ្នាក្នុងចំណោមតម្លៃជាច្រើន។ តម្លៃរបស់វាជាចំនួនថេរ។ វាមានទម្រង់ទូទៅដូចខាង​ ក្រោមនេះ

switch (expression) {

case constan1:

block statement;

break;

case constan2:

block statement;

break;

………….

………….

case constan N:

block statement;

break;

default:

statement;

}

ឧទាហរណ៏

switch ( input\_ch) {

case ‘R’ : cout << “ Red \n” ;

case ‘W’ : cout << “ White \n “;

case ‘B’ : cout << “ Blue \n”;

}

យើងឃើញថា code ខាងលើនេះបើសិនជាវាផ្ទៀតផ្ទាត់នឹងល័ក្ខខ័ណ្ឌទីមួយនោះវានឹងបង្ហាញ តម្លៃដែលវាផ្ទៀងផ្ទាត់រហូតមកដល់ក្រោម។ ដូចនេះ ដើម្បីខ័ណ្ឌដាច់ពី case មួយទៅ case មួយទៀត យើងត្រូវប្រើឃ្លា break ។

ឧទាហរណ៏ទី១

switch (input\_ch) {

case ‘A’ :

case ‘a‘ : cout << “ A “ ; break;

case ‘R’ :

case ‘r‘ : cout << “ R “ ; break;

case ‘P’ :

case ‘p‘ : cout << “ P “ ; break;

case ‘M’ :

case ‘m‘ : cout << “ M “ ; break;

default : cout << “ Not a vowel \n “;

}

ឧទាហរណ៏ទី២

#include<iostream.h>

void main() {

int day;

cout << “ Enter a number between 1 to 7 \n”;

cin >> day ;

switch (day) {

case 1: cout << “ Monday \n” ; break;

case 2: cout << “ Tuesday \n” ; break;

case 3: cout << “ Wednesday \n “ ; break;

case 4: cout << “ Thursday \n “; break;

case 5: cout << “ Friday \n” ; break;

case 6: cout << “ Saturday \n” ; break;

case 7: coutt << “ Sunday \n” ; break;

}

}

ឧទាហរណ៏ទី៣

#include<iostream.h>

void main() {

char op;

float num1, num2, answer;

cout << “ Enter an arithmetic expression “;

cin >> num1 >> op >> num2 ;

switch ( op) {

case ‘+’ :

answer = num1 + num2;

cout << num1 << “+” << num2 << “ = “<< answer << endl;

break;

case ‘-’ :

answer = num1 - num2;

cout << num1 << “-” << num2 << “ = “<< answer << endl;

break;

case ‘\*’ :

answer = num1 \* num2;

cout << num1 << “\*” << num2 << “ = “<< answer << endl;

break;

case ‘/’ :

answer = num1 / num2;

cout << num1 << “/” << num2 << “ = “<< answer << endl;

break;

default:

cout << “ Invalid operator “<< endl;

}

}

២. ទម្រង់ដំណើរការប្រតិបត្តិដដែលៗឬ Loops

២.១ ការប្រើ for loop

ទម្រង់​ for loop មានបីកន្សោម។ កន្សោមទីមួយត្រូវបានប្រើសម្រាប់កំណត់តម្លៃដំបូង កន្សោមទី២ ប្រើសម្រាប់ផ្ទៀងផ្ទាត់ថាតើ loop នឹងត្រូវបតន្តម្តងទៀតឬទេ ហើយកន្សោមទី៣ សម្រាប់ប្តូរតម្លៃដើម្បីប្រតិបត្តិបន្ត។ ទម្រង់របស់​ for loop គឺ

for ( expression1; expression2; expression3)

Statement;

ឧទាហរណ៏

Int sum = 0;

for (int I = 1; I < = 100; i++)

sum = sum +1;

អញ្ញាតដែលប្រើក្នុង​ loop ពេលនេះគឺជា​ i ។ នៅក្នុងជំហានទី១អញ្ញាតនេះត្រូវបាន កំណត់តម្លៃដំបូងស្មើ 1 ។​ បន្ទាប់មកកន្សោមទី២ ត្រូវរង្វាយតម្លៃ ហើយបើសិនជាល័ក្ខខ័ណ្ឌ ពិតនោះវាត្រូវប្រតិបត្តិក្នុង loop ។ ក្រោយពីឃ្លាទាំងឡាយនៅក្នុង loop បានប្រតិបត្តិការ ពេល នោះកន្សោមទីបីត្រូវរង្វាយតម្លៃ។ លទ្ធផលនឹងបង្កើនតម្លៃអោយ i ចំនួន 1 ឯកតា។

ឧទាហរណ៏ទី១

#include<iostream.h>

void main() {

for (int i= 0; i < = 10 ; i++)

cout << i << ‘\t’ ;

}

ឧទាហរណ៏ទី២

#include<iostream.h>

void main() {

int n;

cout << “ How many numbers ? ”;

cin >> n;

float a , sum = 0, avg;

for (int I = 0; i <=n-1; i++) {

cout << “Enter a number “;

cin >> n;

sum = sum +a;

}

Avg = sum/n;

cout << “ sum = “ << sum << endl;

cout << “ Average = “ << avg << endl;

}

២.២ ការប្រើ while loop

While loop ត្រូវបាបប្រើនៅពេលយើងមិនប្រាកដថានឹងត្រូវធ្វើការប្រតិបត្តិ។ បន្ទាប់ពី បានពិនិត្យឃើញថាល័ក្ខខ័ណ្ឌដំបូងមានតម្លៃពិតហើយនោះ វានឹងចូលទៅធ្វើការប្រតិបត្តិនៅ ក្នុង lopp តែម្តង។

ទម្រង់ទូទៅនៃ while loop គឺ

while (condition)

statement;

ទម្រង់នេះពុំបានបង្ហាញអោយឃើញច្បាស់អំពីការកំណត់តម្លៃដំបូង ឬផ្នែកកំណើន តម្លៃរបស់ loop ឡើយ។ ដូចនេះដើម្បីសរសេរទំរង់ពេញលេញរបស់វា គេត្រូវសរសេរដូច​ខាង ក្រោយនេះ។

Initial condition

while (condition) {

statement1;

statement2;

change of the initial condition;

}

ឧទាហរណ៏ទី១

Sum =1 + 2 + 3 + …+15

#include<iostream.h>

void main() {

int sum = 0;

int i=1;

while(i<=15)

{

Sum += i;

i++;

}

cout <<” 1+ 2+ 3+…+15 =” << sum << endl;

}

ឧទាហរណ៏ទី២

#include<iostream.h>

void main() {

int n, i;

float avg, a ,sum=0;

cout << “ How many numbers ?\n”;

cin >> n;

i=0;

while( i < = n-1) {

cout << “ Enter a number : “;

cin >> a;

sum = sum +a;

i++;

}

avg=sum/n;

cout << “ sum = “ << sum << ‘\t’;

cout << “ Average =” << avg << endl;

}

២.៣ ការប្រើ do-while

do – while មានការប្រើស្រដៀងទៅនឹង while loop ដែរតែវាចាបើផ្តើមដំបូង ចូលទៅប្រតិបត្តិក្នុង loop ជាមុនសិន មុននឹងផ្ទៀងផ្ទាត់ល័ក្ខខ័ណ្ឌ។ ដូចនេះវាប្រតិបត្តិការយ៉ាងហោច ណាស់ក៏បានមួយដងដែរ។ បើសិនល័ក្ខខ័ណ្ឌពិតវាចេះតែធ្វើសកម្មភាពដដែលៗនៅក្នុង loop រហូតដល់ល័ក្ខខ័ណ្ឌមិនពិត។ វាមានទម្រង់ទូទៅដូចខាងក្រោមនេះ

initial expression

do {

statement1;

statement2;

…………….

Change of the initial expression;

}while(condition);

ឧទាហរណ៏ទី១

sum = 2 + 4 + 6+ …. + n

#include<iostream.h>

void main() {

int n , sum=0 , i=2, a;

cout << “ Enter a number : \n”;

cin >> n;

do {

cout << “ Enter a number : “;

cin >> a;

sum = sum + i;

i = i + 2;

}while(I < = n);

Cout << “ 2 + 4 + 6 + …. + = “<< n << “ sum = “<< sum << endl;

}

ឧទាហរណ៏ទី២

#include<iostream.h>

void main() {

int n,i;

float avg, a ,sum = 0;

cout << “ How many numbers? “;

cin >> n;

i=0;

do {

cout << “ Enter a number =”;

cin >> a;

sum = sum +a;

}while (i <= n-1);

Avg = sum/n;

cout << “ sum = “ << sum << endl;

cout << “ Average = “ << avg << endl;

}

ឧទាហរណ៏ទី៣

#include<iostream.h>

#include<conio.h>

void main() {

char ch;

float dis,pay,total;

do {

clrscr();

cout << “Enter pay= “ ;

cin >> pay;

if (pay < 300)

dis= pay\*0.03;

else if ( pay < 500)

dis = pay \* 0.05;

else

dis = pay \* 0.07;

total = pay -dis;

cout << “ Pay = “ << pay << endl << “Dis = “<< dis << endl;

cout << “ Total = “ << total <<endl;

cout << “ Press y to continue, Esc to Exit “;

ch =getch();

}while(ch==’y’ || ch == ‘Y’);

៣ ការប្រើឃ្លាបំបែកចេញពីន័ក្ខខ័ណ្ឌ

៣.១ ការប្រើឃ្លា break;

ឃ្លា break ត្រូវបានគេប្រើសម្រាប់បញ្ចប់ល័ក្ខខ័ណ្ឌពី loop នៃទម្រង់​ case-switch ។ លើសពីនេះវាអាចប្រើនៅកន្លែងផ្សេងៗដូចជាក្នុង while loop ​ជាដើមក្រោមន័ក្ខខ័ណ្ឌណា​ ​ មួយ។

ឧទាហរណ៏ទី១

//using break statement inside the while loop

#include<iostream.h>

void main() {

int value, i = 0;

while ( i < = 10) {

cout << “ Enter a number: “;

cin >> value;

if ( value < = 0 ) {

cout << “ Zero or negative value found “;

break;

}

i++;

}

}

ឧទាហរណ៏ទី២

// break loop example

#include<iostream.h>

void main() {

int n;

for ( n = 10 ; n > 0 ; n--) {

cout << n << “ , “ ;

if( n==3)

cout << “countdown aborted! “;

break;

}

}

}

៣.២ ការប្រើឃ្លា continue

ឃ្លា continue ប្រើសម្រាប់អនុវត្តសកម្មភាពបន្តនូវការប្រតិបត្តិដូចគ្នាម្តងទៀត ទោះជាវាប្រទះនឹង error ក៏ដោយ។

ឧទាហរណ៏ទី១

//using continue statement

#include<iostream.h>

void main() {

int value , i ;

i = 0;

while ( i <= 10) {

cout << “ Enter a number : “;

cin >> value;

if ( value < = 0) {

cout << “ Zero or negative value found “ ;

continue;

}

i++;

}

}

ឧទាហរណ៏ទី២

#include<iostream.h>

void main() {

for ( int n = 10 ; n > 0; n--) {

if ( n == 5 )

continue;

cout << n << “ , “ ;

}

cout << “ Good Luck !”;

}

៣.៣ ការប្រើឃ្លា goto

ឃ្លា goto ត្រូវបានប្រើសម្រាប់ផ្លាស់ប្តូរការប្រតិបត្តិកម្មវិធី ដោយលោតរំលងទៅផ្នែក ណាមួយផ្សេងទៀតនៃកម្មវិធី។

វាមានទម្រង់ទូទៅ goto label;

គេអាចប្រើឃ្លា goto ក្នុងលក្ខណ:ពីរយ៉ាងគឺ goto តាមល័ក្ខខ័ណ្ឌ (conditional goto) និង goto មិនអាស្រ័យល័ក្ខខ័ណ្ឌ (unconditional goto )​។

ឧទាហរណ៏ទី១

//unconditional goto statement

#include<iostream.h>

void main() {

start:

cout << “ The C++ Programming Language\n”;

goto start;

}

ឧទាហរណ៏ទី២

// conditional goto statement

#include<iostream.h>

void main() {

int a, b ;

cout << “ Enter two numbers : ” ;

cin >> a >> b;

if ( a > b)

goto output1;

else

goto output2;

output1:

cout << “ Largest value = “ << a << endl;

goto stop:

output2:

cout << “ Largest value = “<< b << endl;

stop:

}

ឧទាហរណ៏ទី៣

//Magic number program

#include<iostream.h>

#include<stdlib.h>

void main() {

int option;

int magic;

magic =rand();

do {

cout << “ 1. Get a new magic number\n”;

cout << “ 2. Play\n”;

cout << “ 3. Quit\n”;

do {

cout << “ Enter your choice : “;

cin >> option;

}while(option < 1 || option > 3);

switch ( option) {

case 1:

magic =rand();

break;

case 2:

int t, x;

for (t=0; t < 100 ; t++) {

cout << “Guess the number: “;

cin >> x;

if (x == magic) {

cout << “ \*\* Right \*\* “;

break;

}

else if (x < magic)

cout << “ Too low \n” ;

else

cout << “ Too high.\n “ ;

}

Cout << “ You’ve used up all your guesses. Try again \n “;

break;

case 3:

cout << “ Goodbye\n “ ; break;

}

}while(option!=3);

}