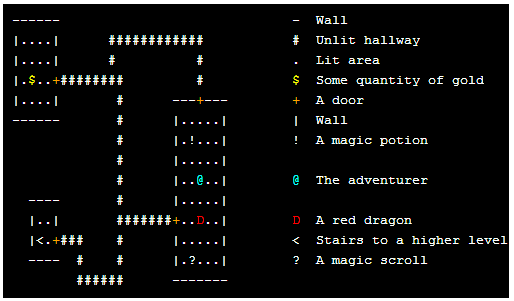
**OOP in C++ : အပိုင္း (၁၃) - Nested if...else Statements**

ဒီအပတ္ သင္ခန္းစာမွာေတာ႔ **Nested if...else Statements** အေၾကာင္းေလးေတြကို ဆက္လက္ ေဆြးေႏြးသြားမွာ ျဖစ္ပါတယ္။

**Nested if...else Statements**

****

ပံု(၁၃-၁) Text-based game Roguelike

ကၽြန္ေတာ္တို႔ ငယ္ငယ္က Microsoft Windows ေတြ မေပၚေသးပါဘူး။ MS-DOS system ကိုပဲ သံုးၾကရပါတယ္။ အဲဒီေခတ္တုန္းက character-mode မွာ ကစားခဲ႔ရတဲ႔ text-based adventure စြန္႔စားခန္း ဂိမ္းေလးေတြကို အမွတ္ရမိပါေသးတယ္။ စိတ္ကူးယဥ္ ကမာၻတစ္ခုထဲမွာ ကၽြန္ေတာ္တို႔ character ေလးတစ္လံုးကို အရပ္ေလးမ်က္ႏွာ လွည္႔ပတ္သြားေစရင္းနဲ႔ ရဲတိုက္ေတြ၊ ေမွာ္ဆရာေတြ၊ ရတနာေတြ အစရွိသျဖင္႔ ရွာေဖြရတာမ်ိဳးပါ။ ဒါေပမယ္႔ အဲဒီဂိမ္းေတြထဲမွာ ရုပ္ပံုေတြ မပါေသးပါဘူး။ အဂၤလိပ္ အကၡရာေလးေတြ ကိုပဲ ပံု(၁၃-၁)မွာ ျပထားသလိုမ်ိဳး input နဲ႔ output အတြက္ သံုးခဲ႔ၾကရတာပါ။ ငယ္ရြယ္ေသးတဲ႔ စာဖတ္သူမ်ား အဖို႔ေတာ႔ အဲဒီလို ကြန္ပ်ဴတာ ေခတ္ဦးက ဂိမ္းေလးေတြကို ေတြ႔ဖူးဖို႔ မလြယ္ေတာ႔ပါဘူး။ ဒီေခတ္မွာေတာ႔ 3D graphics အေကာင္းစားေတြနဲ႔ ဂိမ္းအလန္းစားေတြ က အစားထိုး ေနရာယူ ထားလိုက္ၾကၿပီေလ။ ဒီသင္ခန္းစာမွာေတာ႔ Nested if…else statement အေၾကာင္းကို ရွင္းျပဖို႔အတြက္ အဲဒီလို စြန္႔စားခန္း ဂိမ္းေလးတစ္ခုရဲ႕ အေျခခံ အစိတ္အပိုင္းတစ္ခုကို တုပ ေရးသားျပထားပါတယ္။ adifelse.cpp ကို ေအာက္ပါအတိုင္း ေလ႔လာၾကည္႔ၾကပါစို႔-

// adifelse.cpp

// demonstrates IF...ELSE with adventure program

#include <iostream>

using namespace std;

#include <conio.h> //for getche()

int main()

{

char dir=’a’;

int x=10, y=10;

cout << “Type Enter to quit\n”;

while( dir != ‘\r’ ) //until Enter is typed

{

cout << “\nYour location is “ << x << “, “ << y;

cout << “\nPress direction key (n, s, e, w): “;

dir = getche(); //get character

if( dir==’n’) //go north

y--;

else

if( dir==’s’ ) //go south

y++;

else

if( dir==’e’ ) //go east

x++;

else

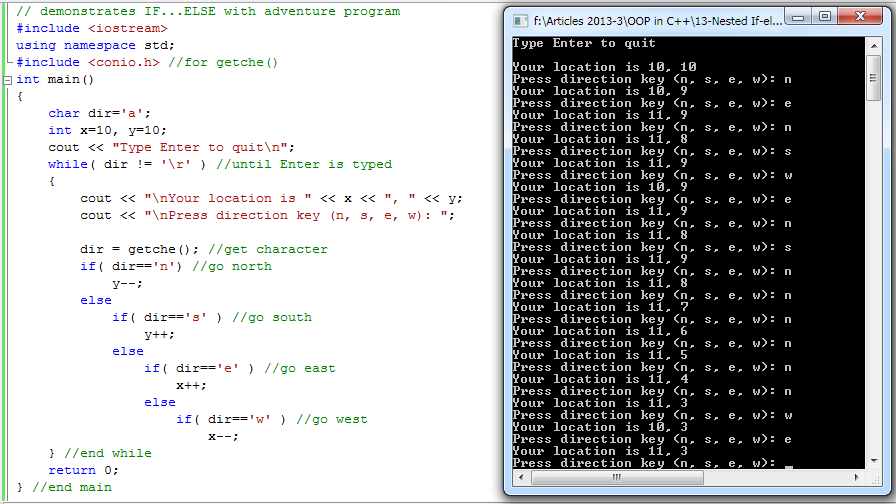
if( dir==’w’ ) //go west

x--;

} //end while

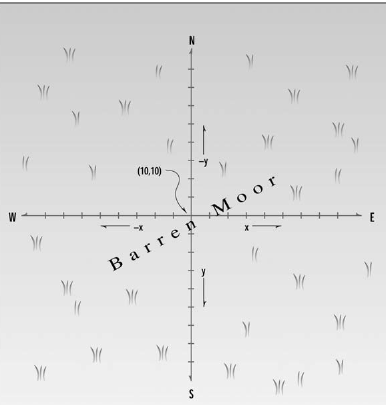
return 0;

} //end main



ပံု(၁၃-၁) Nested if…else အသံုးျပဳပံုကို ေရးသားစမ္းသပ္ျပထားပံု

ဒီဂိမ္းရဲ႕ အစမွာ ကၽြန္ေတာ္တို႔အေနနဲ႔ လြင္တီးေခါင္ တစ္ခုကို ေရာက္ေနသလို ခံစားရမွာ ျဖစ္ပါတယ္။ တစ္ႀကိမ္မွာ အရပ္ေလးမ်က္ႏွာထဲက ကိုယ္ႀကိဳက္တဲ႔ အရပ္ကို တစ္ယူနစ္ သြားလို႔ ရမွာ ျဖစ္ပါတယ္။ပရိုဂရမ္ကေတာ႔ ကၽြန္ေတာ္တို႔ ေရာက္ေနတဲ႔ ေနရာကို x,y coordinate နဲ႔ ျပေပးေနမွာပါ။ စစခ်င္း တည္ေနရာကေတာ႔ (10,10) ျဖစ္ပါတယ္။ ဒီပရိုဂရမ္ေလးမွာေတာ႔ ကၽြန္ေတာ္တို႔ ဘယ္ကိုပဲ သြားသြား ဘာမွ ေတြ႔ရဦးမွာ မဟုတ္ပါဘူး။ တည္ေနရာရဲ႕ x,y တန္ဖိုးေတြပဲ ေျပာင္းလဲ သြားမွာပါ။ ကၽြန္ေတာ္တို႔ အေနနဲ႔ ကန္႔သတ္ခ်က္ ေပးမထားတဲ႔အတြက္ ႀကိဳက္တဲ႔ အရပ္မ်က္ႏွာကို ႀကိဳက္သေလာက္ သြားခြင္႔ ရွိပါတယ္။ အထက္က ပံု (၁၃-၁) မွာ user ထည္႔ေပးလိုက္တဲ႔ အရပ္မ်က္ႏွာကို မူတည္ၿပီး xy coordinate ေတြ ေျပာင္းလဲသြားတာကို ျပထားပါတယ္။ ပရိုဂရမ္ကို အဆံုးသတ္ဖို႔႔ Enter key ကို ႏွိပ္လိုက္ရမွာ ျဖစ္ပါတယ္။ ဒီပရိုဂရမ္ေလးက တကယ္ေတာ႔ game တစ္ခုရဲ႕ အစိတ္အပိုင္း ေသးေသးေလး တစ္ခုပဲ ျဖစ္ပါတယ္။ ဒါေပမယ္႔ multiple branches ေတြကို ကိုင္တြယ္ဖို႔ နည္းလမ္းတစ္ခုကို ေလ႔လာႏိုင္ပါတယ္။ if…else statement တစ္ခုထဲမွာ if…else statement တစ္ခု၊ အဲဒီထဲမွာ ေနာက္ထပ္ if…else statement တစ္ခု နဲ႔ ေနာက္ဆံုးမွာ if statement တစ္ခုကို အဆင္႔ဆင္႔ ထည္႔သြင္းေရးသားထားတာ ျဖစ္ပါတယ္။ ဒါေၾကာင္႔ ပထမဆံုး test condition ဟာ မွားသြားခဲ႔မယ္ဆိုရင္ ဒုတိယ၊ တတိယ စသျဖင္႔ ေလးဆင္႔ စလံုး အဆင္႔ဆင္႔ စစ္ေဆး သြားမွာ ျဖစ္ပါတယ္။ တစ္ခုခု မွန္သြားခဲ႔မယ္ဆိုရင္ေတာ႔ သက္ဆိုင္ရာ x-y တန္ဖိုးမ်ားကို ေျပာင္းလဲေပးတဲ႔ ကုဒ္မ်ားကို လုပ္ေဆာင္ေပးမွာပါ။ အဲဒီလို nested group လုပ္ထားတဲ႔ if…else statements ေတြကို *decision tree* လို႔ ေခၚဆိုၾကပါတယ္။



**Matching the else**

Nested if…else ကို သံုးတဲ႔ေနရာမွာ မွားတတ္တဲ႔ ေနရာေလး တစ္ခု ရွိပါတယ္။ ဒါကေတာ႔ else ကို မွားယြင္းၿပီး မသက္ဆိုင္တဲ႔ if နဲ႔ တြဲသံုးမိတာပဲ ျဖစ္ပါတယ္။ ေအာက္က badelse.cpp ကို ေလ႔လာၾကည္႔ရေအာင္ -

// badelse.cpp

// demonstrates ELSE matched with wrong IF

#include <iostream>

using namespace std;

int main()

{

int a, b, c;

cout << “Enter three numbers, a, b, and c:\n”;

cin >> a >> b >> c;

if( a==b )

if( b==c )

cout << “a, b, and c are the same\n”;

else

cout << “a and b are different\n”;

return 0;

}

အထက္ပါ ပရိုဂရမ္ေလးမွာ cin ကိုသံုးၿပီး a,b နဲ႔ c တန္ဖိုးေတြကို ဖတ္ယူပါတယ္။ user က တန္ဖိုး သံုးခုကို ထည္႔ေပးၿပီး Enter ေခါက္ေပးရမွာပါ။ တကယ္လို႔ ကၽြန္ေတာ္တို႔က 2,3 နဲ႔ 3 ကို ထည္႔ေပးလိုက္တယ္ ဆိုၾကပါစို႔။ ဒါဆို a က 2 b က 3 နဲ႔ c က 3 အသီးသီး ျဖစ္သြားပါၿပီ။ a နဲ႔ b ဟာ မတူတဲ႔ အတြက္ ပထမ if ရဲ႕ test expression ဟာ မွားသြားပါတယ္။ ဒါေၾကာင္႔ ကၽြန္ေတာ္တို႔ အေနနဲ႔ else ကို ေခၚယူၿပီး cout << “a and b are different\n”; ဆိုတဲ႔ ကုဒ္ကို လုပ္ေဆာင္ဖို႔ ေမွ်ာ္လင္႔ထားမိမွာပါ။ ဒါေပမယ္႔ လက္ေတြ႔မွာ ဘာမွ print မထုတ္ေပးပါဘူး။ ဘာလို႔လဲဆိုေတာ႔ အဲဒီ else ကို မသက္ဆိုင္တဲ႔ if နဲ႔ မွားၿပီး တြဲထားခဲ႔မိလို႔ပါပဲ။ ရုတ္တရက္ ၾကည္႔လိုက္ရင္ else က ပထမ if နဲ႔ တြဲထားတယ္လို႔ ထင္စရာ ျဖစ္ေနပါတယ္။ တကယ္က indentation ေနရာခ်ထားမႈအရ အဲလို အထင္မွားေစတာပါ။ တကယ္က အဲဒီ else ဟာ ဒုတိယ if နဲ႔ တြဲေနတာပါ။ ဥပေဒသ ကေတာ႔ else ဟာ ေနာက္ဆံုး if မွာ သာ else မရွိခဲ႔ရင္ ၄င္းနဲ႔ တြဲေနတာ ျဖစ္ပါတယ္။ ေအာက္မွာ indentation နဲ႔ print လုပ္မယ္႔ စာသား ျပင္ထားတဲ႔ ပံုစံ အမွန္ကို ေရးျပထားပါတယ္။

if(a==b)

if(b==c)

cout << “a, b, and c are the same\n”;

else

cout << “b and c are different\n”;

ဒါေၾကာင္႔ 2,3,3 ကို ထည္႔ေပးလိုက္မယ္ ဆိုရင္ ဘာမွ print မထုတ္ေပးေပမယ္႔ 2,2,3 ကို ထည္႔ေပးလိုက္ရင္ေတာ႔ b and c are different ဆိုတာကို print ထုတ္ေပးမွာ ျဖစ္ပါတယ္။ ကၽြန္ေတာ္တို႔ အေနနဲ႔ else ကို ပထမဆံုး if နဲ႔ မျဖစ္မေန တြဲေပးခ်င္ရင္ေတာ႔ ဒုတိယ if ကို ေအာက္ပါအတိုင္း တြန္႔ကြင္းထဲ ထည္႔ေပးလိုက္ရမွာ ျဖစ္ပါတယ္။

if(a==b)

{

if(b==c)

cout << “a, b, and c are the same”;

}

else

cout << “a and b are different”;

ဒီနည္းနဲ႔ else ဟာ ပထမ if နဲ႔ တြဲသြားပါတယ္။ တြန္႔ကြင္း ခတ္ခံထားရတဲ႔ if ကို အျပင္က else က မျမင္ႏိုင္လို႔ပဲ ျဖစ္ပါတယ္။

**The else...if Construction**

if…else ေတြကို nested လုပ္တဲ႔အခါမွာ နားလည္ရ ခက္ေလ႔ရွိပါတယ္။ အထူးသျဖင္႔ ရႈပ္ရႈပ္ေထြးေထြး အဆင္႔အမ်ားႀကီး nested လုပ္ခဲ႔မိရင္ ဖတ္တဲ႔သူအေနနဲ႔ ဘာသာျပန္ဖို႔ ခက္လာပါတယ္။ ဒီျပႆနာကို ေျဖရွင္းဖို႔ ေနာက္တစ္နည္း သံုးႏိုင္ပါတယ္။ ေအာက္ပါ adelseif.cpp ပရိုဂရမ္ေလးကို ေလ႔လာၾကည္႔ၾကရေအာင္ -

// adelseif.cpp

// demonstrates ELSE...IF with adventure program

#include <iostream>

using namespace std;

#include <conio.h> //for getche()

int main()

{

char dir=’a’;

int x=10, y=10;

cout << “Type Enter to quit\n”;

while( dir != ‘\r’ ) //until Enter is typed

{

cout << “\nYour location is “ << x << “, “ << y;

cout << “\nPress direction key (n, s, e, w): “;

dir = getche(); //get character

if( dir==’n’) //go north

y--;

else if( dir==’s’ ) //go south

y++;

else if( dir==’e’ ) //go east

x++;

else if( dir==’w’ ) //go west

x--;

} //end while

return 0;

} //end main

The compiler sees this as identical to ADIFELSE, but we’ve rearranged the ifs so they directly follow the elses. The result looks almost like a new keyword: else if. The program goes down the ladder of else ifs until one of the test expressions is true. It then executes the following statement and exits from the ladder. This format is clearer and easier to follow than the if...else approach.

တကယ္ေတာ႔ အထက္ပါ ကုဒ္ေလးေတြရဲ႕ ရလာဒ္ဟာ adifelse.cpp နဲ႔ အတူတူပါပဲ။ if ေတြကို else ေတြ ေနာက္ကို ေနရာေရႊ႕ေပးလိုက္တာပဲ ကြာပါတယ္။ ရုတ္တရက္ ၾကည္႔လိုက္ရင္ else if ဆိုတဲ႔ keyword အသစ္တစ္ခု သံုးထားတယ္လို႔ေတာင္ ထင္ရပါတယ္။ ဒီလိုေရးသားနည္းဟာ if…else ထက္ ဖတ္ရတာေရာ ေရးသားရတာပါ သိသိသာသာ လြယ္ကူေစပါတယ္။ ဒါေၾကာင္႔ decision tree ပံုစံမ်ိဳး ေရးသားစရာ ၾကံဳလာမယ္ဆိုရင္ if…else ပံုစံထက္ else if ကို အသံုးျပဳသင္႔ေၾကာင္း အၾကံျပဳလိုပါတယ္။

ေနာက္အပတ္ေတြမွာေတာ႔ ေရြးခ်ယ္စရာ အမ်ားအျပားမွ တစ္ခုကို အလြယ္တကု ေရြးထုတ္ေပးႏိုင္တဲ႔ switch statement အေၾကာင္းကို ဆက္လက္ ေဆြးေႏြးသြားပါမယ္ခင္ဗ်ာ။ ဤအခန္းဆက္ ေဆာင္းပါးမ်ားဟာ IT ေက်ာင္းသားမ်ားအတြက္ အထူးရည္ရြယ္ၿပီး ေရးသားထားတာ ျဖစ္ပါတယ္။ သင္ရိုးကို ေရးသားျခင္း ျဖစ္တဲ႔အတြက္ တိုက္ရိုက္ ဘာသာျပန္ဆိုျခင္းနည္းကို ေနရာေတာ္ေတာ္မ်ားမ်ားမွာ သံုးထားပါတယ္။ သိလိုသည္မ်ားကို aungwh2013@gmail.com ကို ဆက္သြယ္ၿပီး ေမးျမန္းႏိုင္သလို YCC ေက်ာင္းသားမ်ား အတြက္လည္း ေအာက္ေဖာ္ျပပါ လိပ္စာရွိ ေတာ္ဝင္ ကြန္ပ်ဴတာ စင္တာ၌လည္း ေလ႔လာ စံုစမ္းႏိုင္ပါတယ္ခင္ဗ်ာ။

**အကိုးအကား**

Object-Oriented Programming in C++(4th edition), Robert Lafore, Copyright©2002 by Sams Publishing: ISBN 0-672-32308-7

Dr. ေအာင္ဝင္းထြဋ္ (bluephoenix)

http://www.tech4mm.com

ေတာ္ဝင္ ကြန္ပ်ဴတာ စင္တာ

၁၇၉ စ၊ သုမဂၤလာ၊ ေစ်းေလး အေနာက္ဘက္၊ ျပင္ဦးလြင္ၿမိဳ႕