**OOP in C++ : အပိုင္း (၁၄) - The switch Statement**

ဒီအပတ္ သင္ခန္းစာမွာေတာ႔ switch statement အေၾကာင္းကို ရွင္းျပေပးသြားမွာ ျဖစ္ပါတယ္။ ကၽြန္ေတာ္တို႔အေနနဲ႔ decision tree အႀကီးစားတစ္ခုကို ေျဖရွင္းဖို႔ လိုအပ္လာတဲ႔ အခါမ်ိဳးမွာ၊ ေနာက္ၿပီး အဲဒီလို ဆံုးျဖတ္ဖို႔ကို variable တစ္ခုတည္းရဲ႕ တန္ဖိုးေပၚ မူတည္ေနမယ္ ဆိုရင္ေတာ႔ if…else ဒါမွမဟုတ္ else if ေတြ အဆင္႔ဆင္႔ ေရးမယ္႔ အစား switch statement ကို အသံုးျပဳသင္႔ပါတယ္။

// platters.cpp

// demonstrates SWITCH statement

#include <iostream>

using namespace std;

int main()

{

int speed; //turntable speed

cout << “\nEnter 33, 45, or 78: “;

cin >> speed; //user enters speed

switch(speed) //selection based on speed

{

case 33: //user entered 33

cout << “LP album\n”;

break;

case 45: //user entered 45

cout << “Single selection\n”;

break;

case 78: //user entered 78

cout << “Obsolete format\n”;

break;

}

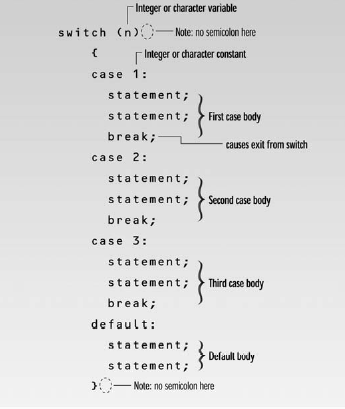
return 0;

}



ဒီပရိုဂရမ္ေလးဟာ တကယ္ေတာ႔ ဓာတ္ျပားေဟာင္းႀကီးမ်ားကို ဖြင္႔တဲ႔ ဓာတ္စက္ေတြရဲ႕ လည္ပတ္မႈ အျမန္ႏႈန္းကို ေရြးခ်ယ္ဖို႔ ေရးထားတာ ျဖစ္ပါတယ္။ user ထည္႔ေပးလိုက္ႏိုင္တဲ႔ 33, 45, နဲ႔ 78 ဆိုတဲ႔ ျဖစ္ႏိုင္ေျခသံုးခုအတြက္ သက္ဆိုင္ရာ messages ေတြကို print ထုတ္ေပးမွာ ျဖစ္ပါတယ္။ အဲဒီေခတ္က ဓာတ္ျပားေတြမွာ long-playing records (LPs) ေတြမွာ သီခ်င္းေတြ အမ်ားႀကီး ပါဝင္ၿပီး 33rpm (တစ္မိနစ္ကို ၃၃ပတ္ႏႈန္း) နဲ႔ လည္ပါတယ္၊ သီခ်င္းတစ္ပုဒ္စီသာပါဝင္တဲ႔ single song ေတြကိုေတာ႔ 45rpm နဲ႔ လည္ပါတယ္။ 78 ကေတာ႔ သူတို႔ LPs နဲ႔ single တို႔ထက္ ေရွးက်တဲ႔ စနစ္ေဟာင္း လည္ပတ္ႏႈန္းပဲ ျဖစ္ပါတယ္။

switch ဆိုတဲ႔ keyword ေနာက္က လက္သဲကြင္း အဖြင္႔အပိတ္ထဲမွာ variable ကို ထည္႔ေပးရမွာ ျဖစ္ပါတယ္ ( ဥပမာ- **switch(speed)** )။ တြန္႔ကြင္း အဖြင္႔အပိတ္ထဲမွာေတာ႔ case statements ေတြကို ေရးသားရမွာပါ။ case တစ္ခုစီရဲ႕ ေနာက္မွာ constant တစ္ခု ပါဝင္ရမွာ ျဖစ္ပါတယ္။ အဲဒီ constant ကို ကြင္းမခတ္ထားဘဲ ေနာက္က colon(**:**) နဲ႔ ပိတ္ေပးရမွာပါ (ဥပမာ - **case 33:**) ။ case constant မွာ အသံုးျပဳထားတဲ႔ constants ေတြဟာ switch variable နဲ႔ အမ်ိဳးအစားခ်င္း တူညီဖို႔ေတာ႔ လိုအပ္မွာ ျဖစ္ပါတယ္။ ေအာက္ကပံုမွာ switch statement ရဲ႕ syntex ကို ျပသထားပါတယ္။



switch မတိုင္ခင္မွာ switch variable ကို တန္ဖိုးတစ္ခု သတ္မွတ္ထားေပးဖို႔ လိုပါတယ္။ အဲဒီတန္ဖိုးကို case statement ေတြမွာ ရွိတဲ႔ constant တန္ဖိုးေတြနဲ႔ တူရာကို ရွာရမွာပါ။ ရည္ရြယ္ထားတဲ႔ case ကိုေတြ႔ၿပီဆိုရင္ အဲဒီေနာက္က statements ေတြကို break မေတြ႔မခ်င္း လုပ္ေဆာင္ေပးမွာ ျဖစ္ပါတယ္။ ေအာက္မွာ platter.cpp ကို run ျပထားပါတယ္။ ေလ႔လာၾကည္႔ၾကရေအာင္ -

Enter 33, 45, or 78: 45

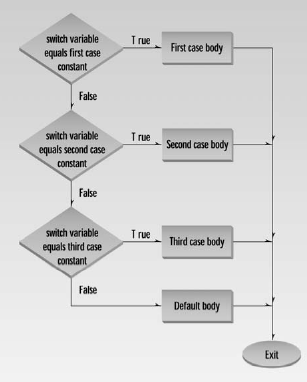
Single selection

**The break Statement**

**platters.cpp** ရဲ႕ case section အားလံုးရဲ႕ အဆံုးမွာ break statement ေရးသားထားတာကို သတိထားမိမွာပါ။ တကယ္ေတာ႔ break keyword ဟာ switch statement တစ္ခုလံုးကေန ထြက္သြားေစမွာပါ။ ပရိုဂရမ္ရဲ႕ control ဟာ break ကိုေတြ႔ရင္ switch ရဲ႕ အျပင္ဘက္ကို ထြက္သြားၿပီး သူ႔ေနာက္က statement ကို ဆက္လက္ လုပ္ေဆာင္မွာ ျဖစ္ပါတယ္။

break ကိုေရးဖို႔ မေမ႔သင္႔ပါဘူး။ break မပါခဲ႔ရင္ ေနာက္ထပ္ case ေတြကို ဆက္လက္ စစ္ေဆးေနဦးမွာပါ။ break statement ကို loop ေတြထဲက ထြက္ခ်င္တဲ႔ အခါမ်ိဳးမွာလဲ အသံုးျပဳႏိုင္ပါတယ္။ ေနာက္ပိုင္းၾကံဳလာတဲ႔ အခ်ိန္မွာ ဆက္လက္ ရွင္းျပသြားပါ႔မယ္။

တကယ္လို႔ switch variable ဟာ case constants ေတြနဲ႔ တစ္ခုမွ မတူညီခဲ႔ဘူးဆိုရင္ ဘာမွ မလုပ္ေဆာင္ဘဲ switch ထဲက ထြက္သြားမွာ ျဖစ္ပါတယ္။



**switch Statement with Character Variables**

platters.cpp မွာ အသံုးျပဳခဲ႔တဲ႔ switch variable ဟာ int data type ျဖစ္ပါတယ္။ ကၽြန္ေတာ္တို႔အေနနဲ႔ char data type ကိုလည္း အသံုးျပဳႏိုင္ပါေသးတယ္။ ယခင္ သင္ခန္းစာက adelseif.cpp ကို ျပန္ျပင္ေရးထားတဲ႔ adswitch.cpp ကို ေလ႔လာၾကည္႔ၾကရေအာင္ -

// adswitch.cpp

// demonstrates SWITCH with adventure program

#include <iostream>

using namespace std;

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

int main()

{

char dir=’a’;

int x=10, y=10;

while( dir != ‘\r’ )

{

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

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

dir = getche(); //get character

switch(dir) //switch on it

{

case ‘n’: y--; break; //go north

case ‘s’: y++; break; //go south

case ‘e’: x++; break; //go east

case ‘w’: x--; break; //go west

case ‘\r’: cout << “Exiting\n”; break; //Enter key

default: cout << “Try again\n”; //unknown char

} //end switch

} //end while

return 0;

} //end main

ဒီပရိုဂရမ္ထဲမွာ switch variable အေနနဲ႔ dir ဆိုတဲ႔ character variable ကို အသံုးျပဳထားတာ ျဖစ္ပါတယ္။ char constant ေတြ အေနနဲ႔ကေတာ႔ 'n', 's', 'e', 'w' နဲ႔ '\r' တို႔ကို အသံုးျပဳထားပါတယ္။ ဒီေနရာမွာ '\r' က Enter ကို ကိုယ္စားျပဳထားတဲ႔ character ျဖစ္ပါတယ္။ (တစ္ခု မွတ္သားထားဖို႔က ကၽြန္ေတာ္တို႔အေနနဲ႔ integer နဲ႔ character variable ေတြကို switch variable အေနနဲ႔ အသံုးျပဳလို႔ ရေပမယ္႔ floating-point number ေတြကိုေတာ႔ အသံုးမျပဳႏိုင္ဘူးဆိုတာပါပဲ)။

ဒီပရိုဂရမ္မွာ ကုဒ္ေတြက တိုေတာင္းတာ ျဖစ္တဲ႔အတြက္ case ေနာက္က statements ေတြကို တစ္ေၾကာင္းတည္းမွာပဲ တန္းစီ ေရးလိုက္ျခင္းျဖင္႔ ပိုမို က်စ္လစ္သြားေစပါတယ္။ '\r' ကို သံုးၿပီး Enter ေခါက္ရင္ exit message ေပၚလာေအာင္ ေရးသားထားတာကို ေလ႔လာႏိုင္ပါတယ္။

**The default Keyword**

adswitch.cpp မွာ switch ရဲ႕ case ေတြ အဆံုးမွာ default ဆိုတဲ႔ keyword ပါေနတာကို သတိထားမိမွာပါ။ ၄င္းဟာ switch variable နဲ႔ case constant ေတြ တစ္ခုမွ မကိုက္ညီတဲ႔ အခါမ်ိဳးမွာ တစ္ခုခု လုပ္ေဆာင္ႏိုင္ဖို႔ ထည္႔ထားျခင္း ျဖစ္ပါတယ္။ ဒီ ဥပမာမွာေတာ႔ user က အျခား character တစ္ခုခုကို ထည္႔သြင္းခဲ႔ရင္ Try again! ဆိုတာကို print ထုတ္ေပးျခင္းအားျဖင္႔ ျပင္ဆင္ ထည္႔သြင္းခြင္႔ ေပးထားပါတယ္။ default ဟာ switch ရဲ႕ ေနာက္ဆံုး statement ျဖစ္တာေၾကာင္႔ break ကို ထည္႔ေပးဖို႔ မလိုေတာ႔ပါဘူး။

switch statement ကို user ထည္႔ေပးလိုက္တဲ႔ input ကို စမ္းစစ္ဖို႔ ေယဘုယ် သံုးတတ္ၾကပါတယ္။ အေျခအေနအမ်ားႀကီးထဲက တစ္ခုကို ေရြးခိုင္းတာပဲ ျဖစ္ပါတယ္။ DOS ေခတ္ေတြတုန္းက menu ေရးဖို႔ do-while နဲ႔ switch ကို တြဲသံုးေလ႔ ရွိခဲ႔ပါတယ္။ default statement ကိုေတာ႔ ကၽြန္ေတာ္တို႔ အေနနဲ႔ မလိုအပ္ဘူးလို႔ ထင္ေနရင္ေတာင္မွ မျဖစ္မေန ထည္႔ထားသင္႔ပါတယ္။ default: cout << “Error: incorrect input to switch”; break; ဆိုတဲ႔ ကုဒ္တစ္ေၾကာင္းက ပရိုဂရမ္မာ သို႔မဟုတ္ user ကို input မွားယြင္း ထည္႔ေပးေနၿပီ ဒါမွမဟုတ္ တစ္ခုခုေတာ႔ အမွားအယြင္း ရွိေနၿပီ ဆိုတာကို သတိေပးႏိုင္ပါတယ္။ ပရိုဂရမ္ အေသးေတြမွာ သိပ္မသိသာေပမယ္႔ အေရးႀကီးတဲ႔ ပရိုဂရမ္တိုင္းမွာ အမွားေတြကို တတ္ႏိုင္သမွ် ကုဒ္ေတြနဲ႔ ေထာင္ေခ်ာက္ဆင္ၿပီး ေဖာ္ထုတ္တတ္ဖို႔ လိုအပ္ပါတယ္။

**switch Versus if...else**

ကၽြန္ေတာ္တို႔ အေနနဲ႔ switch နဲ႔ if...else (ဒါမွမဟုတ္ else if) ေတြကို ဘယ္ေနရာမွာ ဘာကို သံုးရမယ္ဆိုတာကို တိတိက်က် ဆံုးျဖတ္ရ ခက္ေနတတ္ပါတယ္။ တကယ္ေတာ႔ else if ကို variable အမ်ိဳးမ်ိဳးနဲ႔ ရႈပ္ရႈပ္ေထြးေထြး သံုးထားတဲ႔ အခါမ်ိဳးမွာ မျဖစ္မေန သံုးသင္႔ပါတယ္။ ဥပမာ -

f( SteamPressure\*Factor > 56 )

// statements

else if( VoltageIn + VoltageOut < 23000)

// statements

else if( day==Thursday )

// statements

else

// statements

switch statement မွာေတာ႔ variable တစ္ခုတည္းကိုပဲ စစ္ေဆးၿပီး ဆံုးျဖတ္တာ ျဖစ္ပါတယ္။ switch မွာေတာ႔ case a<3: // do something break; ဆိုၿပီး ေရးလို႔ မရပါဘူး။ 'a', 3 စတဲ႔ character ဒါမွမဟုတ္ integer constant နဲ႔ constant တန္ဖိုး ထြက္လာမယ္႔ 'a'+3 လို expression မ်ိဳးကိုပဲ သံုးရမွာ ျဖစ္ပါတယ္။

အဲဒီလို အေျခအေနမ်ိဳးနဲ႔ ကိုက္ညီတဲ႔အခ်ိန္မွာ swith statement ဟာ ေရးသားရတာ အင္မတန္ ရွင္းလင္း၊ လြယ္ကူၿပီး ဖတ္ရသူကိုလည္း နားလည္ရ လြယ္ေစပါတယ္။ ဒါေၾကာင္႔ decision tree မ်ားလာတာနဲ႔အမွ် switch ကို အတတ္ႏိုင္ဆံုး အသံုးျပဳေစခ်င္ပါတယ္ ခင္ဗ်ာ။

ေနာက္အပတ္ေတြမွာေတာ႔ conditional operatorမ်ား 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