**Working with if/else and switch statements (a problem to work on in the lab for WEEK#6)**

Develop a functional decomposition and write a C++ program that inputs a two-letter abbreviation for one of the 50 states and prints out the full name of the state. If the abbreviation isn’t valid, the program should print an error message. Here are a few sample names of the 50 states and their abbreviations

**State Abbreviation State Abbreviation**

**Alabama AL California CA  
Alaska AK Colorado CO  
Arizona AR Connecticut CT**

**(Hint: Use nested Switch statements, where the outer statement uses the first letter of the abbreviation as its switch expression.] What is a benefit in using switch statements?**

Something to think about..  
num=4;  
cout <<num++;

num=5;  
cout<<++num;

num=10;  
if(num++ >10)  
 cout<<”num is greater than 10.\n”;

num=10;  
if(++num >10)  
 cout<<”num is greater than 10.\n”;

Forms of the switch Statement

switch (IntegerExpression)  
{  
 case ConstantExpression://place one or more  
 //statements here …make sure to end each   
 // statement with a ;   
 break;  
 case ConstantExpression://place one or more  
 //statements here …make sure to end each   
 // statement with a ;   
 break;  
 //case statements may be repeated as many time  
 //as necessary  
 default:  
 // place one or more statements here  
}   
The controlling expression given in parenthesis after the keyword switch must always return either a bool value, an enum constant, one of the integer types, or a character.

Forms of the if Statement

Simple if Example  
if (expression) if (x < y)  
 statement; x++;

If/else Example  
if (expression) if (x <y)  
 statement; x++;  
else else  
 statement; x--;

If/else if Example  
if (expression) if (x <y)  
 statement; x++;  
else if (expression) else if (x < z)  
 statement; x--;  
else else  
 statement; y++;

To conditionally-execute more than one statement, enclose the statements in braces:

Form Example  
if (expression) if (x < y)  
{ {  
 statement; x++;  
 statement; cout <<x;  
} }

See also page links for wk#5-6 (course downloads page) on (1) relational and logical operators and (2) flowcharting – decision control structures