PARAMETER PASSING BY USING PASS BY REFERENCE TECHNIQUE USING REFERENCE VARIABLE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

program in c++ to illustrate the swapping of two integers

using a function by implementing call by reference using

reference variables

\*/

#include<iostream.h>

#include<conio.h>

void swap(int &,int &);

int main()

{

int a,b;

clrscr();

a=15;

b=20;

cout<<"values of variables before swapping : "<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

swap(a,b);

cout<<"values of variables after swapping : "<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

getch();

return 0;

}

void swap(int &refa, int &refb)

{

int tmp;

tmp=refa;

refa=refb;

refb=tmp;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Remember that the changes made to the reference variable in any function, will be reflected outside that function while accessing any alias of that reference variable. This could be thought of as a strength as well as a weakness or side effect of reference variables.

This property was used as strength in the preceding example, where we needed the changes to be reflected outside the swap() function.

To illustrate the side effects of reference variables we consider the following two programs.

PROGRAM # 1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream.h>

#include<conio.h>

int add(int,int);

int main()

{

int a,b,sum;

a=15;

b=25;

clrscr();

cout<<"the values of variables before calling function : "<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

sum=add(a,b);

cout<<"the values of variables after calling function : "<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

cout<<"sum = "<<sum<<endl;

getch();

return 0;

}

int add(int local\_a,int local\_b)

{

int sum;

sum=local\_a+local\_b;

local\_a++;

local\_b++;

return sum;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROGRAM # 2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream.h>

#include<conio.h>

int add(int&,int&);

int main()

{

int a,b,sum;

a=15;

b=25;

clrscr();

cout<<"the values of variables before calling function : "<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

sum=add(a,b);

cout<<"the values of variables after calling function : "<<endl;

cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

cout<<"sum = "<<sum<<endl;

getch();

return 0;

}

int add(int &ref\_a,int &ref\_b)

{

int sum;

sum=ref\_a+ref\_b;

ref\_a++;

ref\_b++;

return sum;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

In the first program we adopted the pass by value mechanism, in this mechanism the change in values of local variables is not reflected in the other functions, on the other hand, when we consider the pass by reference mechanism using reference variables. And on using the reference variables we found that the change in the reference variables is reflected outside the function which may not be desirable in some cases.