

Exercise # 5 Solution

(01-11-2000)

Exercise 1

State the difference between the following terms:

- a. *Call by value*: A function call in which the formal parameter receives a copy of the contents of the corresponding actual parameter.
Call by reference: A function call in which the formal parameter receives the location (memory address) of the caller's actual parameter
- b. *Scope*: The region of program code where it is legal to reference (use) an identifier.
Lifetime: The period of time during program execution when an identifier has memory allocated to it.
- c. *Local variable*: A variable declared within a block and not accessible outside of that block.
Global variable: Any identifier declared outside of all the functions in a program.

Exercise 2

Given the declarations

```
const int ANGLE = 90;  
char letter;  
int number;
```

Indicate whether each of the following actual parameters would be valid using pass-by-value, pass-by-reference, or both.

	Pass-by-value	Pass-by-reference
a. Letter	*	*
b. ANGLE	*	
c. Number	*	*
d. number + 3	*	
e. 23	*	
f. ANGLE * number	*	
g. abs(number)	*	

Exercise 3

The program below has a function named Change. Fill in the values of all variables before and after the function is called. Then fill in the values of all variables after the return to the main function. (if any value is undefined, write U instead of a number)

```
#include <iostream.h>

void Change(int, int&);

int main ()
{
    int a ;
    int b;
    a = 10;
    b = 7;
    Change(a,b);
    cout << a << ' ' << b << endl;
    return 0;
}

void Change (int x, int& y)
{
    int b;
    b = x;
    y = y + b;
    x = y;
}
```

Variables in main just before Change is called:

a : **10**
b : **7**

Variables in Change at the moment control enters the function:

x : **10**
y : **7**
b : **U**

Variables in main after return from Change:

a : **10**
b : **17**

Exercise 4:

What is the output of the following C++ program? Can you guess which case do the given functions test (local scope, global scope, call by value, call by reference).

```
#include <iostream>

int x;

void func1(int& a)
{
    a = 3;
}

void func2(int b)
{
    b = 4;
}

void func3()
{
    int x;
    x = 5;
}

void func4()
{
    x = 7;
}

int main()
{
    x = 15;
    func1(x);
    cout << x << endl;

    x = 16;
    func2(x);
    cout << x << endl;

    x = 17;
    func3();
    cout << x << endl;

    x = 18;
    func4();
    cout << x << endl;

    return 0;
}
```

3
16
17
7

func1: call by reference

func2: call by value

func3: local scope

func4: global scope

Exercise 5:

For each of the following, decide whether a value-returning function or a void function is the most appropriate implementation

a. Selecting the larger of two values for further processing in an expression.

value-returning function

b. Printing a paycheck.

void function

c. Computing the area of a hexagon.

value-returning function

d. Testing whether an incoming value is valid and returning TRUE if it is.

value-return function

e. Computing the two roots of a quadratic equation.

void function

P.S. Answers at Exercise 5 are most preferred choices. It's of course possible to make all functions as void functions, and use pass by reference to return values, or make all functions as value return functions, and return status of function (function ended correctly, or incorrectly) and perhaps use pass by reference to return value if necessary.