**LAB MANUAL**

**Functions**

**“Passing by reference”**

Programming Fundamentals

BS(DS/AI)



FAST National University of  
Computer and Emerging Sciences

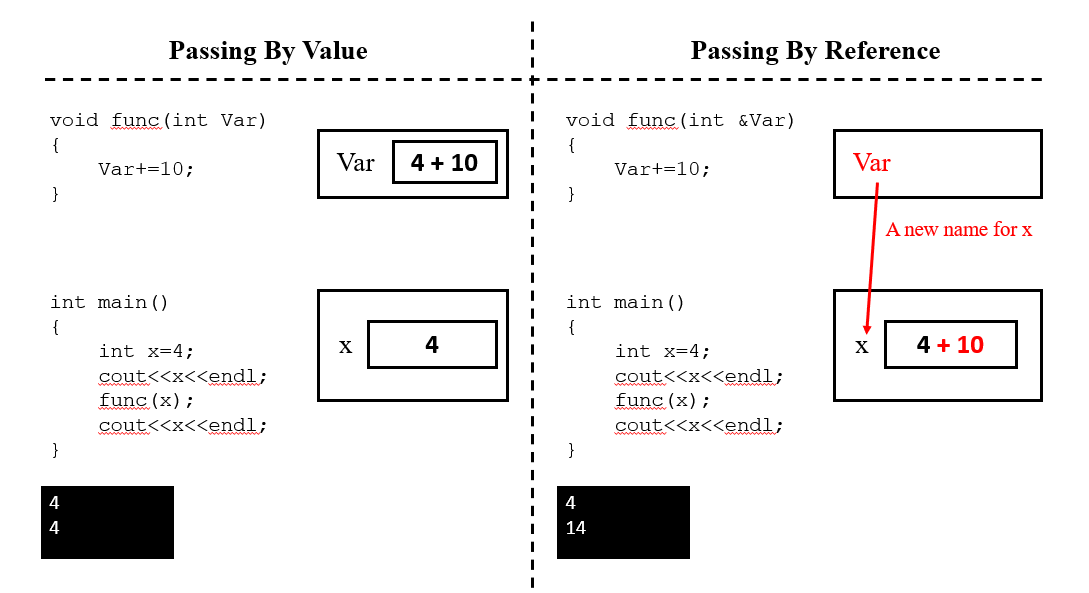
**Difference in Passing by Value and Passing by Reference:**

Passing by Value:

* A variable (suppose its value is 4) is passed to a function,
  + a usual parameter of the function catches the value
  + the function adds 10 to its parameter (parameter’s value)
  + it won’t affect the value of the original variable passed from the main function
  + a separate memory location has already allocated to the original variable in main memory
  + a separate memory location will be occupied by the parameter in the function

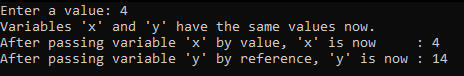
Passing by Reference:

* A variable (suppose its value is 4) is passed to a function,
  + a different parameter is used here this time - address/reference operator followed by parameter name (&param)
  + the function also adds 10 to its parameter
  + it shall change the value of the original variable passed from the main function without using return keyword
  + the parameter in this particular case will act as another, second label (variable) for the same memory location of the variable passed from main function



Example Program #02: Side by side comparison of passing by value and passing by reference for a single integer.

|  |  |
| --- | --- |
| 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | #include<iostream>  **using** **namespace** std;  void byValue(int);  void byReference(int&);  int main()  {  int x, y;  cout<<"Enter a value: ";  cin>>x;  y=x;  cout<<"Variables **\'**x**\'** and **\'**y**\'** have the same values now."<<endl;  byValue(x);  cout<<"After passing variable **\'**x**\'** by value, **\'**x**\'** is now     : "<<x<<endl;  byReference(y);  cout<<"After passing variable **\'**y**\'** by reference, **\'**y**\'** is now : "<<y<<endl;  }  void byValue(int varA)  {  varA+=10;  }  void byReference(int &varB)  {  varB+=10;  } |



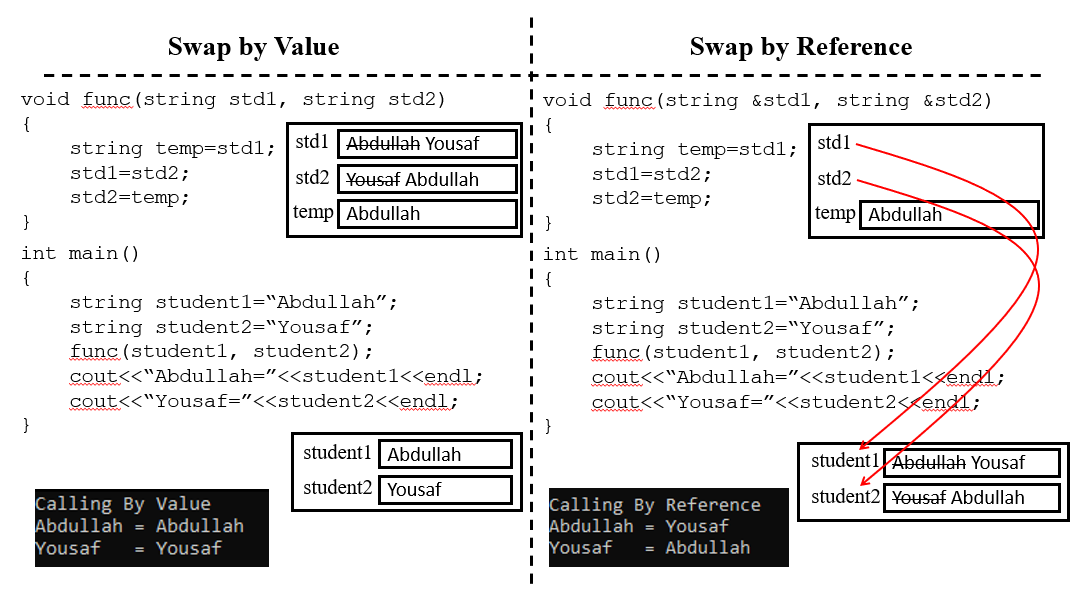
**Swap by Value and Swap by Reference:**

Swap by Value:

* Two strings are passed to a function,
  + both are swapped in the function
  + and, function returns nothing
  + after calling the function and after that, printing the strings in the main function, you will see no change occurred to them

Swap by Reference:

* Two strings are passed to a function by reference (using reference/address operator),
  + both are swapped in the functions
  + and, function returns nothing
  + after calling the function and after that, printing the strings in the main function, you will see that the two strings are actually swapped with each other



Example Program #03: Side by side comparison of passing by value and passing by reference for two strings.

|  |  |
| --- | --- |
| 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | #include<iostream>  #include<string>  **using** **namespace** std;  void swapV(string, string);  void swapR(string&, string&);  int main()  {  string student1 = "Abdullah", student2 = "Yousaf";  cout << "Calling By Value" << endl;  swapV(student1, student2);  cout << "Student1(Abdullah) = " << student1 << endl;  cout << "Student2(Yousaf)   = " << student2 << endl;  cout << "**\n**Calling By Reference" << endl;  swapR(student1, student2);  cout << "Student1(Abdullah) = " << student1 << endl;  cout << "Student2(Yousaf)   = " << student2 << endl;  }  void swapV(string std1, string std2)  {  string temp = std1;  std1 = std2;  std2 = temp;  }  void swapR(string &std1, string &std2)  {  string temp = std1;  std1 = std2;  std2 = temp;  } |

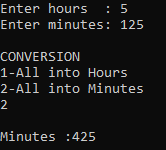
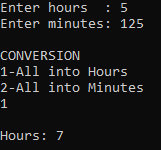
**Lab Tasks:**

**Submission instructions:**

* Submit **single .cpp** file containing code for all patterns.
* Rename your file as **i24XXXX\_LAB06.cpp** and then submit it on GCR.

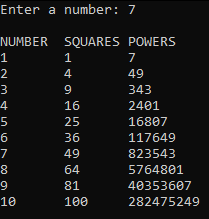
**Task#01:**

* Create two functions,
  + one shall convert hours into minutes – takes hours (by reference) and minutes (by value)
  + other shall convert minutes into hours – takes hours (by value) and minutes (by reference)
* Take the hours and minutes from the user
* Implement the following logic using switch



**Task#02:**

* Create a function (Square()) that can square a variable by reference
* Create another function (Power()) that takes total three parameters
  + two usual parameters, base and exponent
  + and, one reference variable (name this variable result)
  + this function shall find out power of the based as per the exponent and store it in the result
* Take a single integer number from the user
* Now, print the following pattern using any loop where the first column represents the values of the loop



**Task#03:**

Write a program in C++ that contains a function named timesTen . When timesTen is called, it should display the product of number times ten.

* + The program should take number from user.
  + The function takes value of number as an input parameter.
  + The function should calculate ten times of that integer.
  + The function should display product of number times ten