Aror University of Art, Architecture, Design & Heritage Sukkur.

Department of Artificial Intelligence and Multimedia Gaming

Fundamentals of Programming (Fall-2023)

LAB No. 09

Prepared by: Abdul Haseeb Shaikh

Objective of Lab No. 09:

After performing lab 8, students will be able to:

- Use predefined functions
- Write and Use User defined function
- Write functions using function prototype
- Write and use void and return type functions
- Write and Use functions with arrays as parameters

Function:

A function is a group of statements that is executed when it is called from some point of the program. The following is its format:

type name (parameter1, parameter2, ...) {statements}

1. Type is the data type specifier of the data returned by the function.
2. name is the identifier by which it will be possible to call the function.
3. parameters (as many as needed): Each parameter consists of a data type specifier followed by an identifier, like any regular variable declaration (for example: int x) and which acts within the function as a regular local variable. They allow to pass arguments to the function when it is called. The different parameters are separated by commas.
4. Statements is the function's body. It is a block of statements surrounded by braces { }.

Example 1.1

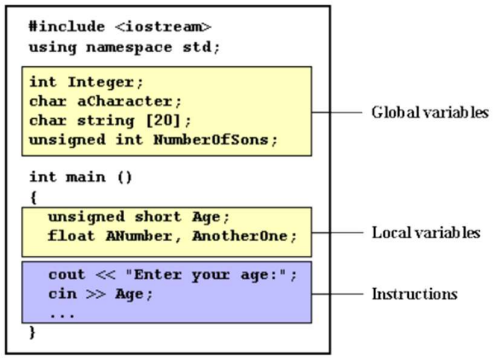
```
// function example
#include <iostream>
using namespace std;

int addition (int a, int b)
{
  int r;
  r=a+b;
  return r;
}

int main ()
{
  int z;
  z = addition (5,3);
  cout << "The result is " << z;
  return 0;
}
```

```
The result is 8
```

Scope of Variables in a function:
The scope of variables declared within a function, or any other inner block is only their own function or their own block and cannot be used outside of them. For example, in the previous example it would have been impossible to use the variables a, b or r directly in function main since they were variables local to function addition. Also, it would have been impossible to use the variable z directly within function addition since this was a variable local to the function main.



Therefore, the scope of local variables is limited to the same block level in which they are declared. Nevertheless, we also have the possibility to declare global variables; These are visible from any point of the code, inside and outside all functions. In order to declare global variables, you simply have to declare the variable outside any function or block; that means, directly in the body of the program.

Functions with no return value- Void Functions:

Imagine that we want to make a function just to show a message on the screen. We do not need it to return any value. In this case we should use the void type specifier for the function. This is a special specifier that indicates absence of type.

```
// void function example
#include <iostream>
using namespace std;

void printmessage ()
{
  cout << "I'm a function!";
}

int main ()
{
  printmessage ();
  return 0;
}
```

```
I'm a function!
```

Default values of parameters:

We can also specify default values for the parameters of a function, so that if one or more arguments after the very first argument are left blank, then values are decided automatically based on the default values specified in function definition.

```
// default values in functions
#include <iostream>
using namespace std;

int divide (int a, int b=2)
{
  int r;
  r=a/b;
  return r;
}
int main ()
{
  cout << divide (12);
  cout << endl;
  cout << divide (20,4);
  return 0;
}
```

```
6
5
```

Pre-defined functions:
Functions which are available in the standard libraries of C++, user do not define those functions, user just call those functions with some specific inputs to get their work done.

```
#include<iostream>
#include<cmath>
using namespace std;

int main(){

cout<<round(1.5);

return 0;
}
```

```
2
--------------------------------
Process exited after 0.06023 seconds with return value 0
Press any key to continue . . .
```

Lab Exercises:

1. **Write down a C++ function, named as count_occurences which takes two parameters: an array and a value, and returns the count of times the value occurs in the array.**

2. **Write a C++ function named merge_text that concatenates two strings. The function should take two strings as parameters and return the concatenated result.**

3. **Implement a function named matrix_mul that performs matrix multiplication. The function should take two matrices as parameters and prints the result of multiplication.**

4. **Implement a function called fact_func to calculate the factorial of a given non-negative integer and returns the factorial.**

5. **Create a function named swap_val that takes two integer parameters and swaps their values. Use this function to swap the values of two variables in the main program.**

6. **Implement a function named power_func that calculates the power of a number. The function should take two parameters, the base, and the exponent.**

7. **Write down a function called sum_max_min, which takes an array as a parameter and returns the sum of maximum and minimum value from the array, this function in turn calls two other functions named as max_array which takes an array and returns maximum value from the array and min_array which takes an array and returns minimum value from the array respectively.**

Lab exercises: