

## Functions in C++

### What is a function?

A function is a block of code or set of statements that takes some input, process it and return some output.

### Why we need functions?

- To reduce code redundancy
- To Break down the complexity
- To promote modularity
- To reuse multiple times by write it once
- For more readability of code
- For maintenance of code
- For Easy to debug

### Function Declaration:

```
Return_type func_name (parameter_list);
```

Return\_type: The data type of the value the function will return.for example int,float etc..and void for no return value.

func\_name: The name we are given to that function

parameter\_list:A comma separated list of variables with their data type that the function accepts as input(It can be empty);

we have to declare a function when we write the function definition after main function

### Function Definition:

```
Return_type func_name (parameter_list){  
    //statements  
}
```

### Function calling:

```
func_name(argument_list);
```

### Example program:

```
#include <iostream>

using namespace std;

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Unsorted array: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    for (int i = 0; i < n - 1; i++) {
        bool swapped = false;
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // Swap elements
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    cout << "\nSorted array: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}
```

Let's write the program using functions:

```
#include <iostream>

using namespace std;

void printArray(int arr[],int n){
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
}
```

```

void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // Swap elements
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Unsorted array: ";
    printArray(arr,n);//to print before sorting

    bubbleSort(arr, n);//to sort the array using bubblesort

    cout << "\nSorted array: ";
    printArray(arr,n);//to print after sorting

    return 0;
}

```

### Output:

PS E:\C++ PROGRAMMES> g++ arraySorting.cpp

PS E:\C++ PROGRAMMES> ./a.exe

Unsorted array: 64 34 25 12 22 11 90

Sorted array: 11 12 22 25 34 64 90

## Home Work:

### 1. Write a function to print counting from 1 to 100

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

void printNumbers(int n){
    for(int i=1;i<=n;i++){
        cout<<i<<" ";
    }
}

int main(){
    int n;
    cout<<"Enter the n value to print numbers:";
    cin>>n;
    printNumbers(n);
}
```

#### Output :

PS E:\C++ PROGRAMMES> g++ printNumbers.cpp

PS E:\C++ PROGRAMMES> ./a.exe

Enter the n value to print numbers:100

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 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53  
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78  
79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

### 2. write a function to find simple interest

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

double simpleInterest(double principal,double rate,double time){
    double interest = (principal * rate * time) / 100.0;
    return interest;
}

int main(){
    double principal, rate, time, interest;
    cout << "Enter principal amount: ";
    cin >> principal;
    cout << "Enter interest rate (%): ";
    cin >> rate;
    cout << "Enter time in years: ";
    cin >> time;
```

```

    interest = simpleInterest(principal,rate,time);

    cout<<"The simple interest is :"<<interest;
}

```

### Output :

Enter principal amount: 1000

Enter interest rate (%): 2

Enter time in years: 2.5

The simple interest is :50

### 3.Writing a function to print prime numbers from 1 to 100

```

#include<iostream>
using namespace std;
void primeRange(){
    int chk;
    for(int i=1;i<=100;i++){
        chk=0;
        for(int j=2;j<i;j++){
            if(i%j==0){
                chk++;
                break;
            }
        }
        if(chk==0&&i!=1){
            cout<<i<<" ";
        }
    }
    cout<<endl;
}
int main(){

    cout<<"Prime numbers between 1 to 100: ";
    primeRange();
    return 0;
}

```

### Output:

PS E:\C++ PROGRAMMES> g++ primeRange.cpp

PS E:\C++ PROGRAMMES> ./a.exe

Prime numbers between 1 to 100: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71  
73 79 83 89 97

#### 4. Write a function check whether someone is eligible for voting or not

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

void eligibility(int age){
    if(age>=18){
        cout<<"you are eligible to vote"<<endl;
    }
    else{
        cout<<"you are not eligible to vote"<<endl;
    }
}

int main(){
    int age;
    cout<<"Enter you age to check voting eligibility: ";
    cin>>age;
    eligibility(age);
}
```

#### Output:

Enter you age to check voting eligibility: 59

you are eligible to vote

#### 5. Write a SIP calculator using function's concept

```
#include <iostream>
#include<math.h>
using namespace std;

double calculateSIP(double monthlyInvestment, double interestRate, int years)
{
    double monthlyInterestRate = interestRate / 1200;
    int totalMonths = years * 12;

    double futureValue = monthlyInvestment * ((pow(1 + monthlyInterestRate,
totalMonths) - 1) / monthlyInterestRate);

    return futureValue;
}

int main() {
    double monthlyInvestment, interestRate;
    int years;
```

```
cout << "Enter monthly investment amount: ";
cin >> monthlyInvestment;

cout << "Enter annual interest rate (%): ";
cin >> interestRate;

cout << "Enter investment duration in years: ";
cin >> years;

double futureValue = calculateSIP(monthlyInvestment, interestRate, years);

cout << "Future value of your SIP: " << futureValue << endl;

return 0;
}
```

### Output:

Enter monthly investment amount: 1000

Enter annual interest rate (%): 1

Enter investment duration in years: 3

Future value of your SIP: 36530