

Experiment No-09: User Defined Functions.

Objectives

- Familiar with different types of user-defined functions.
- Solve various problems using functions.

Function Type - 1: No arguments passed and no return value.

Example: Write a C++ program to demonstrate the use of function type-1.

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

void addition()
{
    int num1,num2, sum; // variable declaration

    cout<<"Enter number 1: "<<endl;
    cin>> num1;
    cout<<"Enter number 2: "<<endl;
    cin>> num2;

    sum = num1 + num2;
    cout<<"Output: "<<sum;
}

int main()
{
    addition(); //Calling the function here

    return 0;
}
```

Function Type - 2: No arguments passed but a return value.

Example: Write a C++ program to demonstrate the use of function type-2.

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

int addition()
{
    int num1,num2, sum; // variable declaration

    cout<<"Enter number 1: "<<endl;
    cin>> num1;
```

```
cout<<"Enter number 2: "<<endl;
cin>> num2;

sum = num1 + num2;
return sum;
}

int main()
{
int s;
s = addition(); //Calling the function here

cout<<"Output: "<< s;

return 0;
}
```

Function Type - 3: Argument passed but no return value.

Example: Write a C++ program to demonstrate the use of function type-3.

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

void addition(int num1,int num2)
{
int sum;
sum = num1 + num2;
cout<<"Output: "<< sum;
}

int main()
{
int var1, var2;
cout<<"Enter number 1: "<<endl;
cin>> var1;
cout<<"Enter number 2: "<<endl;
cin>> var2;

addition(var1,var2); // calling the function with arguments

return 0;
}
```

Function Type - 4: Argument passed and a return value.

Example: Write a C++ program to demonstrate the use of function type-4.

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

int addition(int num1, int num2)
{
    int sum;
    sum = num1+num2;
    return sum;
}

int main()
{
    int var1, var2;
    cout<<"Enter number 1: "<<endl;
    cin>> var1;
    cout<<"Enter number 2: "<<endl;
    cin>> var2;

    int res = addition(var1, var2); // calling the function
    cout<<"Output: " << res;

    return 0;
}
```

Passing One-dimensional Array to a Function

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

// declare function to display marks
void display(int m[5]) { // take a 1d array as parameter
    cout << "Displaying marks: " << endl;

    // display array elements
    for (int i = 0; i < 5; ++i) {
        cout << "Student " << i + 1 << ": " << m[i] << endl;
    }
}

int main() {
    // declare and initialize an array
    int marks[5] = {88, 76, 90, 61, 69};

    // call display function
    // pass array as argument
    display(marks);

    return 0;
}
}
```

Practice Exercise

1. Write a C++ program to add, subtract, multiply and divide two integers using the following types of user-defined functions -
 - No Argument Passed and No return value
 - No Argument Passed but return a value
 - Argument Passed but no return value
 - Argument passed and returns a value
2. Write a C++ program to find the diameter, circumference, and area of a circle using functions.
3. Write a C++ program to get the largest element of an array using the function.
4. Write a C++ program to find the reverse of an array using the function.
5. Write a C++ program to find the sum of two one-dimensional arrays using the function.

Resources (Link)

[\[Functions in C\]](#)