

LEARN **DSA** WITH C++

WEEK :: 02

LEARN **DSA**
WITH C++

PDF

COURSE INSTRUCTOR BY
ROHIT NEGI

MADE BY-
PRADUM SINGHA

[Check Profile](#)

[My Profile](#)

LEARN DSA WITH C++

WEEK :: 02

DAY: 01

DATE: - 24-04-2023

BINARY TO DECIMAL + FUNCTIONS

Binary	Decimal
1 0 1 0	10

Convert ::

1	0	1	0
1×2^3	1×2^2	1×2^1	0×2^0
8	0	2	0

To Convert **Binary divided by 2** because there are 2 digits 0 and 1.

To Convert **Decimal divided by 10** because there are 10 digits 0 to 9.

Binary to Decimal::

1 0 1 0 first take last digit	$1010 \% 10 \Rightarrow 0 \quad \times 2^0$	<pre>cin>>num, sum = 0; int mul = 1; while(num>0) { int rem= num%10; num = num%10; sum = sum+rem X mul; mul = mul X 2; } cout<<sum;</pre>
1 0 1	$101 \% 10 \Rightarrow 1 \quad \times 2^1$	
1 0	$10 \% 10 \Rightarrow 0 \quad \times 2^2$	
1	$1 \% 10 \Rightarrow 1 \quad \times 2^3$	

#Code::

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

int main()
{
    int num, sum = 0, mul = 1;
    cout<<"Enter Binary Digit: ";
    cin>>num;
```

```

while (num>0)
{
    int rem= num%10;
    num = num/10;
    sum = sum+ rem*mul;
    mul = mul*2;
}
cout<<sum;

return 0;
};

```

Use a **long long Data Type** when entering a long input like (101011110001110).

1-s Complement :
[Details know String lecture]

#Scope ::

code	Global Variable	Local Variable
int sum; int mul=0; Int main() { Int rem = ; } cout<<sum;	Here sum, mul is a Global Variable because any where can exceed .	Here rem is aLocal Variable because rem only excesses inside { }. Whatever variable is declared inside of { } It is included in the Local Variable.

#Break and Continue ::

Break :: It is used to terminate the loop.
Continue:: It is used to execute the next iteration in a particular condition.

Break	Continue
<pre>#include<iostream> using namespace std; int main() { int num; for(int i=1; i<=10; i++) { if(i==6) break; cout<<i<<" "; } return 0; };</pre>	<pre>#include<iostream> using namespace std; int main() { int num; for(int i=1; i<=10; i++) { if(i==6) continue; cout<<i<<" "; } return 0; };</pre>

SWITCH CONDITION

Syntex
<pre>switch(expression) { case 1: cout<<""; case 1: cout<<""; default : cout<<"....."; }</pre>

#Code

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

int main()
{
    int num;
    cout<<"Enter the Num: ";
    cin>>num;

    switch (num)
    {
        case 1: cout<<"Sunday";
        break;
        case 2: cout<<"Munday";
        break;
        case 3: cout<<"Tuesday";
```

```

        break;
        case 4: cout<<"Wednesday";
        break;
        case 5: cout<<"Thursday";
        break;
        case 6: cout<<"Friday";
        break;
        case 7: cout<<"Saturday";
        break;
        default: cout<<"Number not valid";
    }

return 0;
};

```

FUNCTION

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function.

```

#include <iostream>
using namespace std;

// create function
int isEvenOdd(int num)
{
    if (num % 2 == 0)
        return 0;
    else
        return 1;
}

int main()
{
    int num;
    cout << "Enter Num: ";
    cin >> num;
    int n = isEvenOdd(num); // call
function
    if (n == 0)
        cout << "Even";
    else
        cout << "Odd";
    isEvenOdd(num);
    return 0;
};

```

int isEvenOdd (int num)

{
#code- **Function Define**

)
int - return int value
int num - int parameter
void - no return value

Default Argument :

int isEvenOdd(int num =5)
{
}

isEvenOdd(num) - call Function
(num) - Argument

Syntax ::

```

return_type Fun_name (Parameter)
{
    // code
}

```

Exp:: 01: Print Factorial

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

// create function
int printFactorial(int num)
{
    int result = 1;
    for(int i=1; i<=num; i++)
        result = result*i;

    return result;
}

int main()
{
    int num;
    cout<<"Enter Num: ";
    cin>>num;

    int ans =printFactorial (num) ;
    cout<<ans;

    return 0;
};
```

When passing big values like 18, 20, 123, use (**long long**) data type .

HOMEWORK

Exp:: 01 Calculate Average value of Two numbers using Function.

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

// create function
int avgTwoNum (int num1, int num2)
{
    int sum = num1+num2;
    int avg = sum/2;

    return avg;
}

int main()
{
    int num1, num2;
    cout<<"Enter Num1: ";
    cin>>num1;
    cout<<"Enter Num2: ";
    cin>>num2;

    int result =avgTwoNum(num1,num2);
    cout<<result;

    return 0;
};
```

Exp::02 : Find Prime Number.

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

void findPrime(int num)
{
    if (num < 2)
    {
        cout<<"Not Prime";
    }
    for (int i=2; i<num; i++)
    {
        if (num%i ==0)
        {
            cout<<"Not Prime";
            return;
        }
        else
```

```
        cout<<"Prime";  
        return;  
    }  
}  
  
int main()  
{  
    int num;  
    cout<<"Enter Num: ";  
    cin>>num;  
  
    findPrime (num) ;  
  
    return 0;  
};
```

Exp:: 03 Print Fibonacci Series ::

LEARN DSA WITH C++

WEEK :: 02

DAY: 02

DATE: - 25-04-2023

FUNCTIONS + ARRAY

Exp :: 01 :: Print “ Odd” and “Even”.

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

void IsEvenOdd(int num)
{
    if(num%2==0)
        cout<<"Even"<<endl;
    else
        cout<<"Odd"<<endl;
    return;
}

int main()
{
    int num1, num2, num3;
    cin>>num1>>num2>>num3;

    IsEvenOdd(num1);
    IsEvenOdd(num2);
    IsEvenOdd(num3);

    return 0;
};
```

Exp :: 02 :: Add Two numbers 8 and 4.

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

void Add()
{
    cout<<8+4;
}

int main()
{
    Add();
}
```

```
    return 0;
};
```

Exp :: 03 :: Add Two numbers take input User?

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

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

int main()
{
    int num1, num2;
    cin>>num1>>num2;
    int result = AddTwoNum(num1,num2);
    cout<<result;

return 0;
};
```

Exp :: 03 :: Print Table ?

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

void PrintTable(int n)
{
    for(int i=1; i<=10; i++)
        cout<<n*i<<endl;
    return;
}

int main()
{
    int n;
    cin>>n;
    PrintTable(n);

return 0;
};
```

Exp :: 03 :: Print Factorial ?

```

#include<iostream>
using namespace std;

int PrintFactorial(int num)
{
    int total = 1;
    for(int i=1; i<=num; i++)
        total = total*i;
    return total;
}

int main()
{
    int num;
    cin>>num;
    int result = PrintFactorial(num);
    cout<<result;
return 0;
};

```

FUNCTION OVERLOADING

Function overloading or method overloading is the ability to create multiple functions of the same name with different implementations.

```

#include<iostream>
using namespace std;
void Add(int num1, int num2)
{
    int sum = num1 + num2;
    cout<<sum<<" ";
    return;
}
void Add(int num1, int num2, int num3)
{
    int sum = num1 + num2 + num3;
    cout<<sum<<" ";
    return;
}
int main()
{
    Add(20, 30);
}

```

```
Add(10, 10, 10);  
return 0;  
};
```

Function execute according to Argument and Parameter (**Argument = Parameter**).

Call by value:: When pass value using Argument.

ARRAY

Arrays are used to store multiple values in a single variable with the same data type.

Define ::

```
int arr[i];
```

7	6	8	8	4	6
Index: 0	1	2	3	4	5

Define Array::

```
arr[6] = {7, 6, 8, 8, 4, 6};
```

Excese Array ::

```
cout<<arr[i];  
cout<<arr[0];  
cout<<arr[2];  
cout<<arr[3];  
cout<<arr[4];
```

LEARN DSA WITH C++

WEEK :: 02

DAY: 03

DATE: - 26-04-2023

Basic Questions in Array

Array Position ::

$Arr[i] = \text{Base address} + i * \text{size of Variable}$

Exp::01 :: print Arr reverse ::

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

int main()
{
    int arr[5] = {6, 9, 7, 3, 1};
    for(int i=4; i>=0; i--)
        cout<<arr[i]<<" ";

return 0;
};
```

Exp::02 :: Find Largest numbers in Array?

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

int main()
{
    int arr[6] = {6, 2, 4, 9, 7, 8};
    int largest = arr[0];
    for(int i=1; i<6; i++)
    {
        if(arr[i]>largest)
            largest = arr[i];
    }
    cout<<largest;

return 0;
};
```

Exp::03 :: Find Odd numbers in Array?

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

int main()
{
    int arr[8] = {2, 5, 3, -3, 8, 4, 2, -6};
    for(int i=0; i<=8; i++)
    {
        if(arr[i]%2==1 || arr[i]%2==--1)
            cout<<arr[i]<<" ";
    }
    return 0;
};
```

Exp::05 :: Print prime numbers in Array? (Using Function)

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

void prime(int num)
{
    if (num < 2)
        return;

    for (int i = 2; i < num; i++)
    {
        if (num % i == 0)
            return;
    }
    cout << num << " ";
}

int main()
{
    int arr[8] = {2, 3, 7, 1, -11, 8, 13, 12};

    for (int i = 0; i < 8; i++)
    {
        prime(arr[i]);
    }

    return 0;
};
```

Exp::06 :: Print Array **input:** {2, 3, 7, -11, 4} **Output:** {4, 2, 3, 7, 11}

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

int main()
{
    int arr[5] = {2, 3, 7, -11, 4};
    int last_arr = arr[4];

    for(int i=3; i>=0; i--)
    {
        arr[i+1]=arr[i];
    }
    arr[0]=last_arr;

    for(int i = 0; i<5; i++)
        cout<<arr[i]<<" ";

return 0;
};
```

Exp::07 :: Print Array [-No repeat element] **input:** {2, 4, 6, 4, 2, 8, 6} **Output:**{ 8 }

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

int main()
{
    int arr[7] = {2, 4, 6, 4, 2, 8, 6};
    int ans =0;
    for(int i=0; i<7; i++)
        ans = ans^arr[i];        //xor process

    cout<<ans;

return 0;
};
```

HOMEWORK

Exp::08 :: Print smallest number in array?

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

int main()
{
    int arr[8] = {7, 11, -2, 8, 170, -4, 0, 11};
    int small_N = arr[0];

    for(int i=1; i<=7; i++)
    {
        if(small_N>arr[i])
        {
            small_N=arr[i];
        }
    }
    cout<<small_N;

return 0;
};
```

Exp::09 :: Print Even number in array?

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

int main()
{
    int arr[8] = {2, 3, 7, -6, -11, 8, 13, 12};
    for(int i=0; i<8; i++)
    {
        if(arr[i]%2==0)
        {
            cout<<arr[i]<<" ";
        }
    }

return 0;
};
```

Exp::09 :: Print Prime number in array? {Without Function}

Try Yourself

LEARN DSA WITH C++

WEEK :: 02

DAY: 04

DATE: - 27-04-2023

Print Array Element + Sorting Algorithm

Exp::01 :: Print array 5 times?

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

int main()
{
    int arr[5] = {2, 3, 8, 7, 4};

    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
            cout << arr[j] << " ";

        cout << endl;
    }

    return 0;
};
```

Exp::02 :: Print array

```
2 3 8 7 4
2 3 8 7
2 3 8
2 3
2
```

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

int main()
{

    int arr[5] = {2, 3, 8, 7, 4};

    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5-i; j++)
            cout << arr[j] << " ";

        cout << endl;
    }
}
```

```

    }

return 0;
};

```

Exp::03 :: Print array

3 13 12 7 4

13 12 7

12 7 4

7 4

4

```

#include<iostream>
using namespace std;

int main()
{
    int arr[5]= {3, 13, 12, 7, 4};
    for(int i=0; i<5; i++)
    {
        for(int j=i; j<5; j++)
            cout<<arr[j]<<" ";

        cout<<endl;
    }

return 0;
};

```

Exp::04 :: Print array reverse ? **Input:** { 3, 11, 13, 8, 4}; **Output:** {4, 8, 13, 11, 3};

```

#include <iostream>
using namespace std;

int main()
{
    int arr[5] = {3, 11, 13, 8, 4};
    for (int i = 0; i < 5; i++)
    {
        for (int j = 4; j >= 0; j--)
            cout << arr[j] << " ";

        cout << endl;
    }

    return 0;
};

```

Exp::05 :: Print array Input: {3, 13,12, 8, 4};

Output:

4 8 12 13 3

8 12 13 3

12 13 3

13 3

3

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

int main()
{
    int arr[5] ={3, 13,12, 8, 4};

    for(int i=0; i<5; i++)
    {
        for(int j=4-i; j>=0; j--)
            cout<<arr[j]<<" ";

        cout<<endl;
    }

    return 0;
};
```

Exp::06 :: Print array Input: {3, 13,12, 8, 4};

Output:

4 8 12 13 3

4 8 12 13

4 8 12

4 8

4

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

int main()
{
    int arr[5]={3, 13,12, 8, 4};
    for(int i=0; i<5; i++)
    {
        for(int j=4; j>=i;j--)
            cout<<arr[j]<<" ";

        cout<<endl;
    }

    return 0;
};
```

Exp::07 :: Print Duplicate array?

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

int main()
{
    int arr[5] = {2, 2, 3, 3, 4};
    int count[5]={0};

    for(int i=0; i<5; i++)
    {
        count[arr[i]]++;
    }
    for(int i=0; i<5; i++)
    {
        if(count[i]>1)
            cout<<i<<" ";
    }

return 0;
};
```

SORTING

sorting algorithm is an algorithm that puts elements of a list into an order.

Exp::

5	2	8	6
---	---	---	---

Sorting Array:

2	5	6	8
---	---	---	---

Exp::01 :: Print Sorted array? **Input:** {1, 0, 2, 1, 0} **Output:** {0, 0, 1, 1, 2}

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

int main()
{
    int arr[5] = {1, 0, 2, 1, 0};
    int count_zero = 0, count_one = 0, count_two = 0;
    for (int i = 0; i < 5; i++)
    {
        if (arr[i] == 0)
            count_zero++;
        else if (arr[i] == 1)
            count_one++;
        else if (arr[i] == 2)
            count_two++;
    }
    cout<<count_zero<<" ";
    cout<<count_one<<" ";
    cout<<count_two<<" ";
    return 0;
}
```

```
        count_one++;
    else
        count_two++;
}

for (int i = 0; i < count_one; i++)
arr[i]=0;
for (int i = count_one; i < count_zero + count_one; i++)
arr[i]=1;
for (int i = count_zero + count_one; i < 5; i++)
arr[i]=2;

for(int i=0; i<5; i++)
cout<<arr[i]<<" ";

    return 0;
};
```

LEARN DSA WITH C++

WEEK :: 02

DAY: 05

DATE: - 28-04-2023

SORTING ALGORITHM (Selection, Bubble, Insertion)

#Sorting in order:

8	6	7	2	4
---	---	---	---	---

Ascending Order:

2	4	6	7	8
---	---	---	---	---

Descending Order:

8	7	6	4	2
---	---	---	---	---

We can easily find any item/numbers using the sorting Algorithm.

Arrange Ascending Order : **Selection Sort**

Input :: { 2, 7, 8, 3, 5, 1}; Output:: {1, 2, 3, 5, 7, 8};

Explain :

2	7	8	3	5	1
---	---	---	---	---	---

Index	0	1	2	3	4	5
-------	---	---	---	---	---	---

Round 1	0	1	7	8	3	5	2	Check 0-5
Round 2	1	1	2	8	3	5	7	Check 1-5
Round 3	2	1	2	3	8	5	7	Check 2-5
Round 4	3	1	2	3	5	8	7	Check 3-5
Round 5	4	1	2	3	5	7	8	Check 4-5

Swap::

Exchange	4	1
index	0	4

```
int temp = arr[4]
```

```
arr[4] = arr[0]
```

```
arr[0] = temp;
```

Code::

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

int main()
{
    int arr[6] = { 2, 7, 8, 3, 5, 1};
    for(int i=0; i<6; i++)
    {
        int index = i;
        for(int j=i; j<6; j++)
        {
            if(arr[j]<arr[index])
                index=j;
        }

        int temp = arr[index];
        arr[index] = arr[i];
        arr[i] = temp;

        cout<<arr[i]<<" ";
    }

    return 0;
};
```

Selection Sort Easy Accuracy: 64.33% Submissions: 66K+ Points: 2

Explore Job Fair for students & freshers for daily new opportunities. Find A Job Today!

Given an unsorted array of size N, use selection sort to sort arr[] in increasing order.

Example 1:

Input:
N = 5
arr[] = {4, 1, 3, 9, 7}

Output:
1 3 4 7 9

Explanation:
Maintain sorted (in bold) and unsorted subarrays.
Select 1. Array becomes **1** 4 3 9 7.

```

1 // } Driver Code Ends
13 class Solution
14 {
15     public:
16     int select(int arr[], int i)
17     {
18         // code here such that selectionSort() sorts arr[]
19     }
20
21     void selectionSort(int arr[], int n)
22     {
23         //code here
24         for(int i=0; i<n; i++)
25         {
26             int index = i;
27             for(int j=i; j<n; j++)
28             {
29                 if(arr[j]<arr[index])
30                     index=j;
31             }
32
33             int temp = arr[index];
34             arr[index] = arr[i];
35             arr[i] = temp;
36         }
37     }
38 };

```

Custom Input Compile & Run Submit

Bubble Sort

5	3	6	1	2
---	---	---	---	---

Round 1	5	3	6	1	2	Check [(n-1)-i]
Round 2	3	5	1	2	6	[(n-1)-i]
Round 3	3	1	2	5	6	[(n-1)-i]
Round 4	1	2	3	5	6	[(n-1)-i]
Index	0	2	3	4	5	


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

int main()
{
    int arr[5] = {5, 3, 6, 2, 1};

    for (int i = 0; i < 5 - 1; i++)
    {
        for (int j = 0; j < 5 - 1 - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    // print Array

    for(int i=0; i<5; i++)
        cout<<arr[i]<<" ";

    return 0;
};
```

The screenshot shows the GeeksforGeeks website interface. At the top, there's a navigation bar with links to Problems, Courses, Job Fair, Contests, and POTD. The main content area displays the 'Bubble Sort' problem, which is categorized as 'Easy'. It provides an accuracy of 59.33%, over 118K submissions, and 2 points. The problem description states: 'Given an Integer N and a list arr. Sort the array using bubble sort algorithm.' It includes two examples: Example 1 with N=5 and arr=[4, 1, 3, 9, 7], resulting in [1, 3, 4, 7, 9]; and Example 2 with N=10. A C++ code editor is open, showing a solution for Bubble Sort. The code defines a class 'Solution' with a 'bubbleSort' function that sorts an array in ascending order using the bubble sort algorithm. The code is in C++ (g++ 5.4) and has an average time of 15m. At the bottom, there are buttons for 'Custom Input', 'Compile & Run', and 'Submit'.

Insertion Sort

4	6	3	11	7	2
---	---	---	----	---	---

Round 1	4	6	3	11	7	2	Check 1-0
Round 2	3	4	6	11	7	2	2 - 0
Round 3	3	4	6	7	2	11	3 - 0
Round 4	3	4	6	2	7	11	4 - 0
Round 5	2	3	4	6	7	11	5- 0

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

int main()
{
    int arr[6] = {4, 6, 3, 11, 7, 2};

    for(int i=0; i<6-1; i++)
    {
        for(int j=i; j>=0; j--)
        {
            if(arr[j]>arr[j+1])
```

```

        {
            int temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
        }
        else
            break;
    }
};

// print array

for (int i = 0; i < 6; i++)
    cout << arr[i] << " ";

return 0;
};

```

GeeksforGeeks

The screenshot shows the GeeksforGeeks website interface. The top navigation bar includes links for Problems, Courses, Job Fair, Contests, and POTD. The main content area displays the 'Insertion Sort' problem, which is marked as 'Easy' with an accuracy of 66.61% and 92K+ submissions. The problem description states: 'The task is to complete the insert() function which is used to implement Insertion Sort.' Example 1 shows an input array [4, 1, 3, 9, 7] and the output [1 3 4 7 9]. Example 2 is also present but empty. On the right, a C++ code editor shows a solution for the 'insert()' function. The code defines a 'Solution' class with a 'insert()' method that implements the insertion sort algorithm. The code is as follows:

```

1 // Driver Code Ends
2 class Solution
3 {
4 public:
5     void insert(int arr[], int i)
6     {
7         //code here
8     }
9 public:
10    //Function to sort the array using insertion sort algorithm.
11    void insertionSort(int arr[], int n)
12    {
13        //code here
14        for(int i=0; i<n-1; i++)
15        {
16            for(int j=i; j>0; j--)
17            {
18                if(arr[j]>arr[j+1])
19                {
20                    int temp = arr[j];
21                    arr[j] = arr[j+1];
22                    arr[j+1] = temp;
23                }
24                else
25                    break;
26            }
27        }
28    }
29 }
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

At the bottom of the code editor, there are buttons for 'Custom Input', 'Compile & Run', and 'Submit'.

GeeksforGeeks

Find the fine::

```

long long int collection = 0;
/*
if(date%2==1)
{
    for(int i = 0; i<n; i++)

```

```
    {  
        if(car[i]%2==0)  
        {  
            collection = collection + fine[i];  
        }  
    }  
}
```

```
    if(date%2==0)  
    {  
        for(int i=0; i<n; i++)  
        {  
            if(car[i]%2==1)  
            {  
                collection = collection + fine[i];  
            }  
        }  
    } */
```

```
    for(int i=0; i<n; i++)  
    {  
        if(date%2 != car[i]%2)  
            collection = collection + fine[i];  
    }
```

```
    return collection;
```