Software Engineering
Tutorial No 1

SHAH Bhargav
DUDHAGARA Akshay

7 October 2018

1 Hello World

Listing 1: Hello World!

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

int main()
{
    cout << "Hello world!"<<endl;
    return 0;
}
```

2 Variables

2.1 Global and other variables

Global variables are gone be written outside of all functions and its gonna be accessible by every function beneath it since they're gonna be writing at the top most of the time it can be used every function. **Local variables** are going to be assigned inside of the functions and those are going to be only accessible by the function itself.

2.1.1 Elementary functions: Mean, Min, and Max

Listing 2: Max-Min-Mean!

```
#include<iostream>
using namespace std;
/*
 * SHAH Bhargav & DUDHAGARA Akshay
 *
 */

int maximum (int A, int B) //creat a function of maximum value
{
    return(A > B ? A:B); //retun the maximum number
}
int minimum (int A, int B) //create a function of minimum value
```

```
13 {
14     return(A < B ? A:B); //return the minimum number
15 }
16 int mean (int A, int B) //create a function of mean value
17 {
18     return(A + B) / 2.0; //return the mean number
19 }
20 int main()
21 {
22     //int A,B;
23     //cin>> A ;
24     //cin>> B ;
25     cout << "maximum(A,B)==" << maximum(4,5) << endl; //print and call the maximum
           function
26     cout << "minimum(A,B)==" << minimum(5,4) << endl; //print and call the minimum
           function
27     cout << "mean (A,B)==" << mean(10,10) << endl;     // print and call the mean function
28     return 0;
29
30 }
```

3 Combination

3.1 Factorial

Listing 3: Factorial

```
1  #include <iostream>
2  using namespace std;
3
4  //SHAH Bhargav & DHUDHAGARA Akshay
5
6  // use double for taking higher value (we can use integer also)
7  double Factorial(unsigned int n) //creat a function of factorial
8  {
9      if (n > 1)
10         return n * Factorial(n - 1); // n is positive variable
11     else
12         return 1;
13 }
14 int main()
15 {
16     int n;
17     cout << "enter a positive interger: ";
18     cin >> n;
19     cout << "Factorial of " << n << " = " <<Factorial(n) << endl; //call the factorial
           function and print the value
20     return 0;
21
22 }
```

3.2 Number of combinations from a set

Listing 4: Combination-Set

```
1  #include <iostream>
2  using namespace std;
3
4
5  //SHAH Bhargav & DHUDHAGARA Akshay
6
```

```
// use double for taking higher value (we can use integer also)
double Factorial(unsigned int n) //creat a function of factorial
{
    if (n > 1)
        return n * Factorial(n - 1); // n is positive variable
    else
        return 1;
}

double Combination(unsigned int n, unsigned int k) //create a function of combination
{
    // calculation according to binomial coefficient
    double n_factorial = Factorial(n); //get value of n factorial
    double k_factorial = Factorial(k); //get value of k factorial
    double nk_factorial = Factorial(n-k); // eq of binomial coefficient

    return n_factorial / (k_factorial * nk_factorial); //return the value
}

int main()
{

   cout << "combination : " << Combination(49, 6) << endl; // print the value (n = 49 and
        k = 6)
   return 0;

}
```

3.3 Number of combinations with repetitions

Listing 5: Combination with Repetitions

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


//SHAH Bhargav & DUDHAGARA Akshay

// use double for taking higher value (we can use integer also)
double Factorial(unsigned int n) //creat a function of factorial
{
    if (n > 1)
        return n * Factorial(n - 1); // n is positive variable
    else
        return 1;
}
double CombinationsReptitions(unsigned int n, unsigned int k) //function fo Comvinations
    reptitions
{
    double up = Factorial(n + k - 1); //calculating the factorial using n and k
        accourding to given eq
    double down = Factorial(k) * Factorial(n - 1); // calculating the factorial using n
        and k accourding to given eq
    return up / down; //return value
}
int main()
{
    cout << "combination_reptitions:" << CombinationsReptitions(5,6) << endl; //print and
         call the function and return
    return 0;
}
```

3.4 Permutations

Listing 6: Permutations

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

//SHAH Bhargav & DUDHAGARA Akshay

// use double for taking higher value (we can use integer also)
double Factorial(unsigned int n) //creat a function of factorial
{
    if (n > 1)
        return n * Factorial(n - 1); // n is positive variable
    else
        return 1;
}

double Permutations(unsigned int n, unsigned int k) // create a function of Permutation
{
    //as  per the equation
    double Top = Factorial(n);
    double Down = Factorial(n - k);
    return Top / Down;
}

int main()
{
    cout << "permutations value : " << Permutations(54,5) << endl; //call the function
        and print value
    return 0;
}
```

4 List of Fibonacci numbers and its relation with the golden ratio

4.1 List of Fibonacci

Listing 7: Fibonacci

```
#include <iostream>
using namespace std;
/*
    SHAH Bharagv & DUDHAGARA Akshay
    In fibonacci the first two number is o and 1 and each subsequence number is the sum
        of the previous two(t1 and t2)
    n is positive interger of fibonacci sequence

 */

int main()
{
    int n, t1 = 0, t2 = 1, nextTerm = 0;

    cout << "Enter the number of term: ";
    cin >> n; // enter number of value  //here enter the howmany term you want(Ex: 10)

    cout << "Fibonacci Series: ";

    for(int i = 1; i <= n; ++i) // increment of i
    {
        if(i == 1) // check the condition
        {

            cout << " " << t1; // print t1(0)
```

```
25              continue; //acts as goto continue
26          }
27
28          if(i == 2) // check the condition
29          {
30              cout << t2 << " ";
31              continue;
32          }
33          nextTerm = t1 + t2;
34          t1 = t2;
35          t2 = nextTerm;
36
37          cout << nextTerm << " ";
38      }
39      return 0;
40
41  }
```

5 Pascal's triangle

Listing 8: Pascal Triangle

```
1  #include<iostream>
2  using namespace std;
3  /*
4   * Pascal Triagle is a triangualar array of the binomial coefficients
5   *
6   */
7  int main()
8  {
9      int rows;  // positive int of row of pascal
10
11     cout << "Enter number of rows: ";
12     cin >> rows;
13     cout << endl;
14
15     for(int i = 0; i < rows; i++) //looping to print pascal lines
16     {
17         int val = 1; //first number of pascal
18         for(int j = 1; j < (rows - i); j++) //looping to print pascal triangle
19          {
20            cout <<  " ";
21         }
22         for(int k = 0; k <= i; k++) //calculation of tringulation
23         {
24             cout << " " << val;
25             val = val*(i-k)/(k+1);
26         }
27         cout << endl << endl;
28
29     }
30     cout << endl;
31     return 0;
32
33 }
```