Nation University of Sciences and Technology (NUST), Balochistan Campus (NBC)

---

# Assignment # 3

Department:
**Computer Science**

Course Title:
**Fundamentals of Computer programming.**

Course Code:
**CS-110**

## ASSIGNMENT TOPICS:

- Integration.

Submitted to: **Sir Mohsin Raza**

Submitted by: **Muhammad (391855)**

1st Semester,

Session **2021-2025**

**Date of Submission: 01/March /2022**

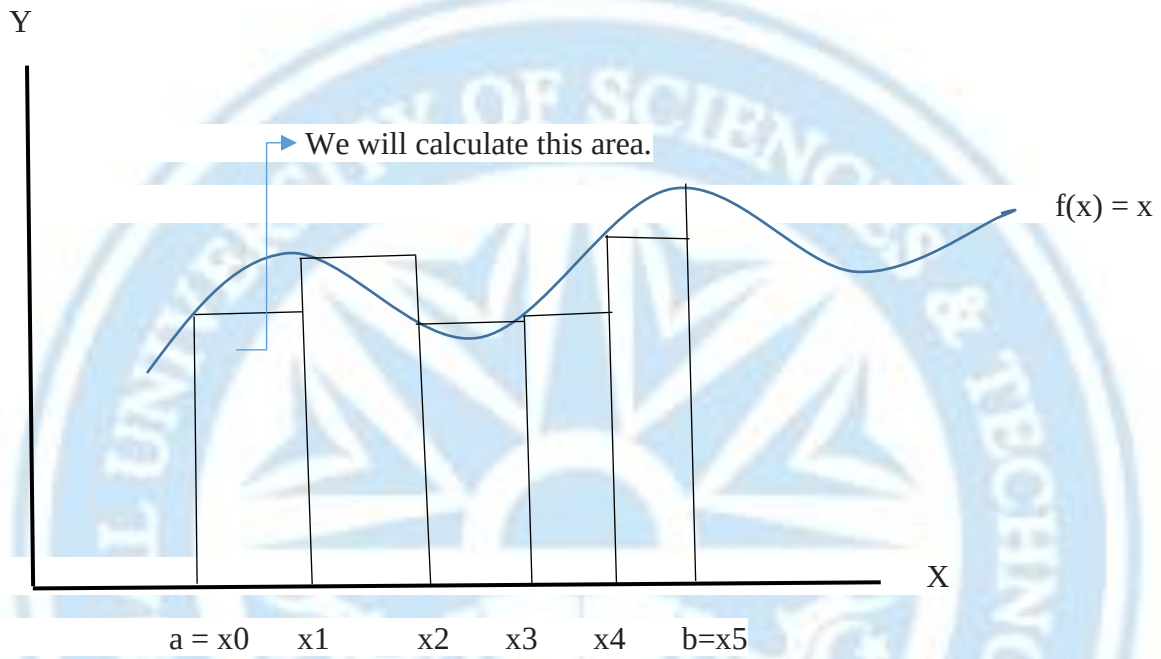
Student sign

Instructor sign

**Calculation of Integral:** Such as this one given below.

$$\int_a^b f(x)dx$$

## The numrical calculation of integal:




- Now we need to calculate the integral b/w two bound, ‘a’ and ‘b’. To do that we numerically divided it into small intervals. As shown in figure above.
- If the regions are in equal spaces, for example the difference b/w x0 and x1 is same as x1 and x2, then that could be the best case for us and simplify our computation.
- The difference b/w x0 and x1, x1 and x2, x2 and x3 and so on, we call it ‘dx’. Like differential of x.
- The area of first rectangle would be ‘the value of function at a’ times ‘dx’.
- The area of Second rectangle would be ‘the value of function at x1’ times ‘dx’. And so on. Then we sum all of these and get the actual approximation.

## The programic calculation of integal:

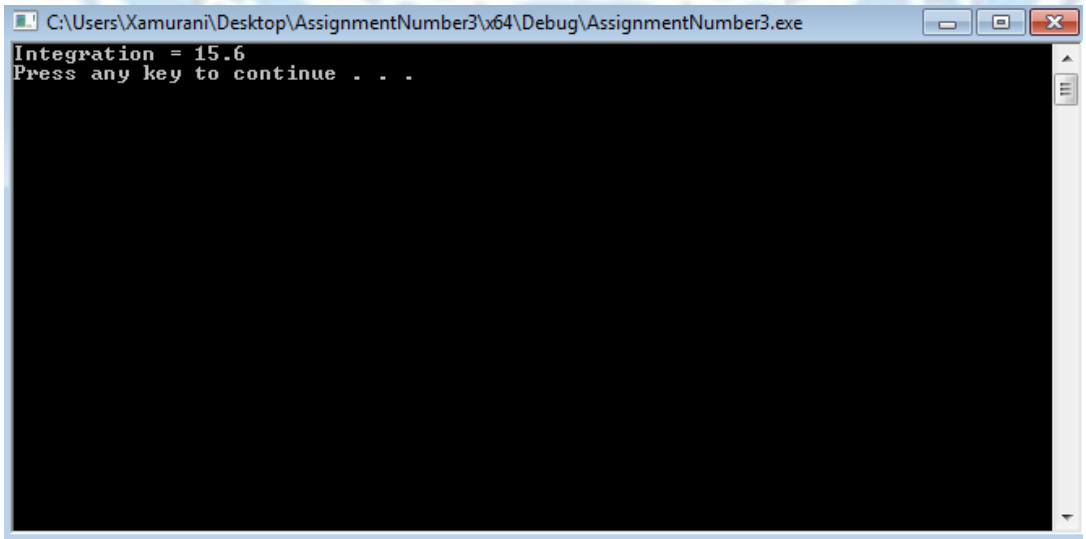
```cpp
#include<iostream>
using namespace std;

double fun(double x)        // It is a funtion of one variable.
{
    return x;
}
// The above function means "f(x) = x".

double sumIntegral(double lowBound, int n, double dx) // User-defined function for integration.
{
    double cumSum = 0;
    for (int i = 0; i < n; i++) {
        double xi = lowBound + i * dx;           // This will give us in ever loop the number x0, x1, x2 and so on.
        double funValue = fun(xi);
        double rectangleArea = funValue * dx;
        cumSum += rectangleArea;
    }
    return cumSum;
}

int main() {
    double lowBound = 4;        // Can also take this from User.
    double upperBound = 7;      // Can also take this from User.
    int n = 5;                // Interval
    double dx = (double)(upperBound-lowBound)/n;   // Taking the derivative.
    double result = sumIntegral(lowBound, n, dx);
    cout << "Integration = " << result << endl;
    system("pause");
    return 0;
}
```

## The Result:




**The End.**