

### Algorithm: Simpson 1/3rd rule

Include predefined class libraries / header files in the program using preprocessor directive #include <iostream> , <cmath>

using namespace std;

**Class declaration:** declare a class "simpson"

**Private member declaration:**

**Data variables:**

declare integer type data variables "i,n"

declare float type data variables "a,b,x[100],y[100],h,so,se,intfx"

where x[100] and y[100] are declared as array and can contain upto 101 values each.

**Public member declaration:**

Declare member function (1) getdata (2) calculate of type void

**Member function definitions:**

Define member functions getdata using scope resolution operator

"void simpson::getdata()"

**Get input from user:**

"enter the value of lower limit of integration a, upper limit of integration b, no of intervals n "

The input is written in allocated memory space using

cin>>a>>b>>n;

**Member function definition:**

Define member functions calculate using scope resolution operator

"void simpson::calculate()"

Perform the calculations using for loop

```
so=0;                //variable initialization
se=0;                //variable initialization
h=(b-a)/n;           //calculate step size

for(i=0;i<=n;i++)    //for loop
{
    x[i]=a+(i*h);     //calculate x[] and y[] for given set of parameters
    y[i]=1/(1+x[i]);
}

for(i=1;i<=n-1;i++)  //for loop
{
    if(i%2==0)        //if loop condition (sum of even terms)
    {
```

```

    se=se+y[i];
}

else                                     //if loop condition (sum of odd terms)
{
    so=so+y[i];
}
}

intfx=(h/3)*(y[0]+(4*so)+(2*se)+y[n]);    //simpson 1/3rd formula
cout<<"intfx="<<intfx;                  //display the result on screen
}

```

Inside the main function of type integer , create an object of class simpson and named s. Use this object to call public member functions and terminate the program using return 0 statement

```

int main()
{
    simpson s;                           //create object
    s.getdata();                          //member function calling
    s.calculate();
    return 0;                             //program termination
}

```