

**Numerical Methods Lab**

**Integration Toolbox Using JAVA**

**Information Technology**

**Section - A**

**Submitted By:**

**NAME & ROLL NO.**

**Hritik Kumar 38**

**Pappu Kumar 40**

**Contents**

1. **Introduction**
   1. **What is Numerical Integration?**
   2. **Element of Numerical Integration**
2. **Methods of Integration**
   1. **Trapezoidal method**
   2. **Simpson 1/3rd Rule**
3. **Source Code**
4. **Output**
5. **Conclusion**

**Introduction**

In mathematics, an **integral** assigns numbers to functions in a way that can describe displacement, area, volume, and other concepts that arise by combining infinitesimal data. Integration is one of the two main operations of calculus, with its inverse operation, differentiation, being the other. Given a function f of a real variable x and an interval [a, b] of the real line, the **definite integral**

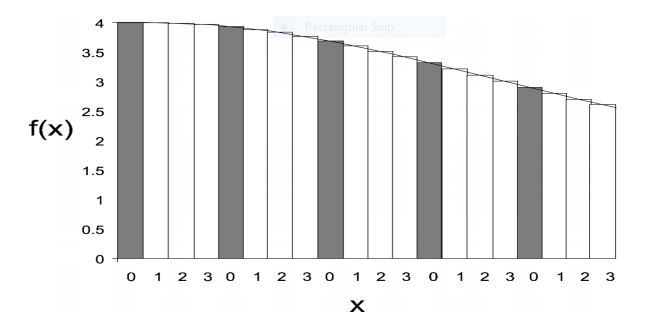
is defined informally as the signed area of the region in the xy-plane that is bounded by the graph of f, the x-axis and the vertical lines x = a and x = b. The area above the x-axis adds to the total and that below the x-axis subtracts from the total. It can be of two types: indefinite integrals and definite integrals.

**What is Numerical Integration ?**

In numerical analysis, numerical integration constitutes a broad family of algorithms for calculating the numerical value of a definite integral, and by extension, the term is also sometimes used to describe the numerical solution of differential equations. The basic problem considered by numerical integration is to compute an approximate solution to a definite integral. It is different from analytical integration in two ways: first it is an approximation and will not yield an exact answer; Error analysis is a very important aspect in numerical integration. Second it does not produce an elementary function with which to determine the area given any arbitrary bounds; it only produces a numerical value representing an approximation of area.

**Elements of Numerical Integration**

If f(x) is a smooth well­behaved function, integrated over a small number of dimensions and the limits of integration are bounded, there are many methods of approximating the integral with arbitrary precision. We consider an indefinite integral:

Numerical integration methods can generally be described as combining evaluations of the integrand to get an approximation to the integral. The integrand is evaluated at a finite set of points called integration points and a weighted sum of these values is used to approximate the integral. For instance if we use rectangles as our shape: 

In this example the definite integral is thus approximated using areas of rectangles. The integration points and weights depend on the specific method used and the accuracy required from the approximation. An important part of the analysis of any numerical integration method is to study the behavior of the approximation error as a function of the number of integrand evaluations. A method which yields a small error for a small number of evaluations is usually considered superior. Reducing the number of evaluations of the integrand reduces the number of arithmetic operations involved, and therefore reduces the total round­off error. Also, each evaluation takes time, and the integrand may be arbitrarily complicated.

**Methods of Integration**

Trapezoidal Rule

The trapezoidal rule works by approximating the region under the graph of the function {\displaystyle f(x)}*f(x)* as a trapezoid and calculating its area. It follows that

=(b-a).(f(a)+f(b))/2

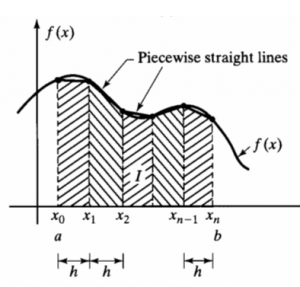
The integral can be even better approximated by partitioning the integration interval, applying the trapezoidal rule to each subinterval, and summing the results. In practice, this "chained" (or "composite") trapezoidal rule is usually what is meant by "integrating with the trapezoidal rule".

C:\Users\acer\Desktop\Capture.JPG

then

C:\Users\acer\Desktop\Capture.JPG

The approximation becomes more accurate as the resolution of the partition increases.When the partition has a regular spacing, as is often the case, the formula can be simplified for calculation efficiency.



{\displaystyle \int \_{a}^{b}f(x)\,dx\approx (b-a)\cdot {\tfrac {f(a)+f(b)}{2}}}

## Error analysis

The error of the composite trapezoidal rule is the difference between the value of the integral and the numerical result:

**C:\Users\acer\Desktop\Capture.JPG**

There exists a number *ξ* between *a* and *b*, such that

**C:\Users\acer\Desktop\Capture.JPG**

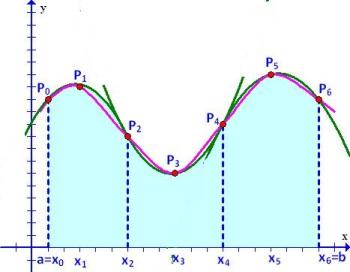
It follows that if the integrand is concave up (and thus has a positive second derivative), then the error is negative and the trapezoidal rule overestimates the true value. This can also be seen from the geometric picture: the trapezoids include all of the area under the curve and extend over it. Similarly, a concave-down function yields an underestimate because area is unaccounted for under the curve, but none is counted above. If the interval of the integral being approximated includes an inflection point, the error is harder to identify.

**Simpson's rule**

**Simpson's rule** is a method for numerical integration, the numerical approximation of definite integrals. Specifically, it is the following approximation for **n** {\displaystyle n}nnnn equally spaced subdivisions

**C:\Users\acer\Desktop\Capture.JPG**

where C:\Users\acer\Desktop\Capture.JPG



## Error analysis

## The error in approximating an integral by Simpson's rule for n=2 is

## C:\Users\acer\Desktop\Capture.JPG

## The error is asymptotically proportional to {\displaystyle (b-a)^{5}}(b-a)5However, the above derivations suggest an error proportional to {\displaystyle (b-a)^{4}}(b-a)4. Simpson's rule gains an extra order because the points at which the integrand is evaluated are distributed symmetrically in the interval {\displaystyle [a,\ b]} [a,b].Since the error term is proportional to the fourth derivative of*f*at  this shows that Simpson's rule provides exact results for any polynomial*f* of degree three or less, since the fourth derivative of such a polynomial is zero at all points.If the second derivtive *f”* {\displaystyle f''} exists and is convex in the interval {\displaystyle (a,\ b)}(a,b):

## *C:\Users\acer\Desktop\Capture.JPG*

**Source Code**

***Main GUI***

package numerical\_calculator;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import net.objecthunter.exp4j.Expression;

import net.objecthunter.exp4j.ExpressionBuilder;

public class MainGUI extends javax.swing.JFrame {

private Object field;

public MainGUI() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

t1 = new javax.swing.JTextField();

t2 = new javax.swing.JTextField();

t3 = new javax.swing.JTextField();

t4 = new javax.swing.JTextField();

t5 = new javax.swing.JTextField();

b1 = new javax.swing.JButton();

tf1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 100, Short.MAX\_VALUE)

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 100, Short.MAX\_VALUE)

);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setFont(new java.awt.Font("Cambria", 1, 60)); // NOI18N

jLabel1.setForeground(new java.awt.Color(153, 0, 153));

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText(" INTEGRATION CALCULATOR");

jLabel2.setFont(new java.awt.Font("Copperplate Gothic Light", 1, 24)); // NOI18N

jLabel2.setForeground(new java.awt.Color(51, 0, 204));

jLabel2.setText("Enter the function to be Integrated:");

jLabel3.setFont(new java.awt.Font("Copperplate Gothic Light", 1, 24)); // NOI18N

jLabel3.setForeground(new java.awt.Color(51, 0, 204));

jLabel3.setText("Integrate w.r.t variable:");

jLabel4.setFont(new java.awt.Font("Copperplate Gothic Light", 1, 24)); // NOI18N

jLabel4.setForeground(new java.awt.Color(51, 0, 204));

jLabel4.setText("No. of Sub-Intervals:");

jLabel5.setFont(new java.awt.Font("Copperplate Gothic Light", 1, 24)); // NOI18N

jLabel5.setForeground(new java.awt.Color(51, 0, 204));

jLabel5.setText("Initial Limit:");

jLabel6.setFont(new java.awt.Font("Copperplate Gothic Light", 1, 24)); // NOI18N

jLabel6.setForeground(new java.awt.Color(51, 0, 204));

jLabel6.setText("Final Limit:");

t1.setFont(new java.awt.Font("Calibri Light", 0, 24)); // NOI18N

t1.setForeground(new java.awt.Color(51, 204, 0));

t1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

t1ActionPerformed(evt);

}

});

t2.setFont(new java.awt.Font("Calibri Light", 0, 24)); // NOI18N

t2.setForeground(new java.awt.Color(51, 204, 0));

t3.setFont(new java.awt.Font("Calibri Light", 0, 24)); // NOI18N

t3.setForeground(new java.awt.Color(51, 204, 0));

t4.setFont(new java.awt.Font("Calibri Light", 0, 24)); // NOI18N

t4.setForeground(new java.awt.Color(51, 204, 0));

t5.setFont(new java.awt.Font("Calibri Light", 0, 24)); // NOI18N

t5.setForeground(new java.awt.Color(51, 204, 0));

t5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

t5ActionPerformed(evt);

}

});

b1.setFont(new java.awt.Font("Century Gothic", 1, 24)); // NOI18N

b1.setForeground(new java.awt.Color(102, 51, 0));

b1.setText("Simpson's 1/3");

b1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b1ActionPerformed(evt);

}

});

tf1.setFont(new java.awt.Font("Nirmala UI Semilight", 0, 30)); // NOI18N

tf1.setForeground(new java.awt.Color(0, 153, 153));

jButton1.setFont(new java.awt.Font("Century Gothic", 1, 24)); // NOI18N

jButton1.setForeground(new java.awt.Color(102, 51, 0));

jButton1.setText("Trapezoidal");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 883, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(42, 42, 42))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(48, 48, 48)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGap(152, 152, 152))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGap(68, 68, 68))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 426, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel5)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 163, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(t4, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(t3, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(t2, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(t5)

.addComponent(t1, javax.swing.GroupLayout.PREFERRED\_SIZE, 269, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(80, 80, 80)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 215, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(147, 147, 147)

.addComponent(b1, javax.swing.GroupLayout.PREFERRED\_SIZE, 215, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(tf1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 767, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 128, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(t1, javax.swing.GroupLayout.DEFAULT\_SIZE, 51, Short.MAX\_VALUE)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(t2, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 51, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(t3, javax.swing.GroupLayout.PREFERRED\_SIZE, 52, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 51, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(t4, javax.swing.GroupLayout.PREFERRED\_SIZE, 54, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(t5, javax.swing.GroupLayout.PREFERRED\_SIZE, 52, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(29, 29, 29)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 74, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b1, javax.swing.GroupLayout.PREFERRED\_SIZE, 74, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 57, Short.MAX\_VALUE)

.addComponent(tf1, javax.swing.GroupLayout.PREFERRED\_SIZE, 84, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(371, 371, 371))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(43, 43, 43)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 20, Short.MAX\_VALUE)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))};

pack();

private void t1ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void t5ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void b1ActionPerformed(java.awt.event.ActionEvent evt) {

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

ExpressionBuilder input= new ExpressionBuilder(t1.getText());

double a=Double.parseDouble(t4.getText());

double b=Double.parseDouble(t5.getText());

int n=Integer.parseInt(t3.getText());

Expression f = input.variable("x").build();

tf1.setText("Integration by Simpson's 1/3 is "+Double.toString(+new Simpson\_one\_third(f,n).eval\_Integral(a,b)));

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

ExpressionBuilder input= new ExpressionBuilder(t1.getText());

double a=Double.parseDouble(t4.getText());

double b=Double.parseDouble(t5.getText());

int n=Integer.parseInt(t3.getText());

Expression f = input.variable("x").build();

tf1.setText("Integration by Trapezoidal is "+Double.toString(+new Trapazoidal(f,n).evaluateIntegral(a,b)));

}

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} java.awt.EventQueue.invokeLater(() -> {

new MainGUI().setVisible(true);

});

}

private javax.swing.JButton b1;

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JTextField t1;

private javax.swing.JTextField t2;

private javax.swing.JTextField t3;

private javax.swing.JTextField t4;

private javax.swing.JTextField t5;

private javax.swing.JTextField tf1;

}

**Simpson’s one – third**

package numerical\_calculator;

import net.objecthunter.exp4j.Expression;

public class Simpson\_one\_third extends AbstractSolver {

public Simpson\_one\_third(Expression f, int n) {

super(f, n);

}

public double eval\_Integral(double a, double b) {

if(a>b)

throw new IllegalArgumentException("INVALID INPUT");

if(n%2!=0)

throw new IllegalArgumentException("INVALID INPUT");

double y =(b-a)/n;

double sum;

sum = (f.setVariable("x", b).evaluate() +f.setVariable("x", a).evaluate())/3;

for(int i = 1; i<n; i++)

if(i % 2 !=0)

sum+= 4\*(f.setVariable("x", a+i\*y).evaluate())/3;

else

sum+= 2\*(f.setVariable("x", a+i\*y).evaluate())/3;

return sum \* y;

}}

**Trapazoidal Rule**

package numerical\_calculator;

import net.objecthunter.exp4j.Expression;

public class Trapazoidal extends AbstractSolver{

public Trapazoidal(Expression f,int n){

super(f,n);

}

public double evaluateIntegral(double a, double b) {

if(a>b)

throw new IllegalArgumentException("INVALID INPUT");

double y =(b-a)/n;

double sum;

sum = (f.setVariable("x", b).evaluate() +f.setVariable("x", a).evaluate())/2;

for(int i=1; i<n; i++)

sum+=f.setVariable("x", a+i\*y).evaluate();

return sum \* y;

}

}

**Abstract Solver**

**package numerical\_calculator;**

**import net.objecthunter.exp4j.Expression;**

**public abstract class AbstractSolver implements IntegrationSolver{**

**protected Expression f;**

**protected int n;**

**public AbstractSolver(Expression f,int n){**

**if(f==null||n<=0)**

**throw new IllegalArgumentException("INVALID INPUT");**

**this.f=f;**

**this.n=n;**

**}**

**public void setN(int n){**

**if(n<=0)**

**throw new IllegalArgumentException("INVALID INPUT");**

**this.n=n;**

**} }**

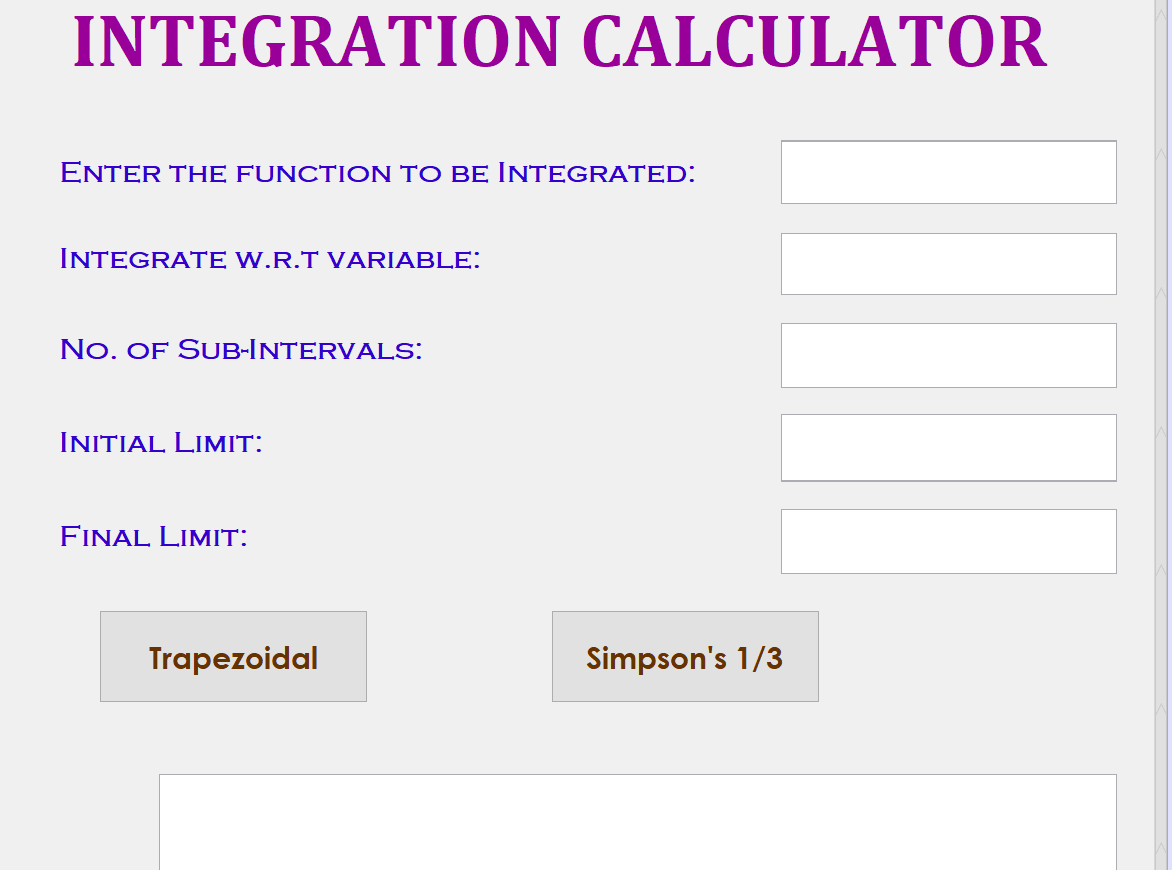
**public interface IntegrationSolver {**

**}**

**public class IllegalArgumentExpression {**

**}**

**Output**

****

**Conclusion**

The Trapezoid Rule and Simpson’s one-third rule is nothing more than the average of the left-hand and right-hand Riemann Sums.

It provides a more accurate approximation of total change than either sum does alone. Simpson’s Rule is a weighted average that results in an even more accurate approximation.

Formula for the Trapezoid rule (replaces function with straight line segments)

Formula for Simpson’s rule (uses parabolas, so exact for quadratics)

Approximations improve as ∆x shrinks

Generally, Simpson’s rule superior to trapezoidal.

We are applying trapezoidal method rule in making our project.