calculator

AIM :

To learn the basis of applet and develop basic calculator with functionalities of addition , subtraction , multiplication and division.

Program :

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class calculator extends Applet implements ActionListener{

Button addition,sub,mul,div,enter,clear;
Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
TextField tf;
TextField tf2;
TextField tf3;
static int a=-1,b=-1,res;

public void init()
{
tf = new TextField();
tf.setBounds(50,50,150,20);


tf2 = new TextField();
tf2.setBounds(50,100,150,20);

tf3 = new TextField();
tf3.setBounds(50,150,150,20);
tf3.setEditable(false);

addition = new Button("+");
addition.setBounds(50,200,60,50);

sub = new Button("-");
sub.setBounds(150,200,60,50);

mul = new Button("*");
mul.setBounds(50,250,60,50);

div = new Button("/");
div.setBounds(150,250,60,50);

b0=new Button("0");
b0.setBounds(220,50,20,20);
add(b0);
b0.addActionListener(this);


b1=new Button("1");
```

```
b1.setBounds(220,70,20,20);
add(b1);
b1.addActionListener(this);

b2=new Button("2");
b2.setBounds(220,90,20,20);
add(b2);
b2.addActionListener(this);

b3=new Button("3");
b3.setBounds(220,110,20,20);
add(b3);
b3.addActionListener(this);


b4=new Button("4");
b4.setBounds(220,130,20,20);
add(b4);
b4.addActionListener(this);

b5=new Button("5");
b5.setBounds(250,50,20,20);
add(b5);
b5.addActionListener(this);

b6=new Button("6");
b6.setBounds(250,70,20,20);
add(b6);
b6.addActionListener(this);

b7=new Button("7");
b7.setBounds(250,90,20,20);
add(b7);
b7.addActionListener(this);

b8=new Button("8");
b8.setBounds(250,110,20,20);
add(b8);
b8.addActionListener(this);

b9=new Button("9");
b9.setBounds(250,130,20,20);
add(b9);
b9.addActionListener(this);

enter=new Button("enter");
enter.setBounds(270,90,40,40);
add(enter);
enter.addActionListener(this);
```

calculator

```
clear=new Button("clear");
clear.setBounds(270,140,40,40);
add(clear);
clear.addActionListener(this);

add(addition);add(sub);add(mul);add(div);
add(tf);
add(tf2);
add(tf3);

addition.addActionListener(this);
sub.addActionListener(this);
mul.addActionListener(this);
div.addActionListener(this);

setSize(500,500);
setLayout(null);
setVisible(true);
}

public void actionPerformed(ActionEvent e)
{
String s1=tf.getText();
String s2=tf2.getText();

/*
int a=Integer.parseInt(s1);
int b=Integer.parseInt(s2);
int res=0;

*/

if(e.getSource()==b0)
{
if(b==-1)
{
b=0;
}
else
{
b=b*10;
}

}

if(e.getSource()==b1)
{
if(b==-1)
b=1;
else
b=b*10+1;
}

if(e.getSource()==b2)
{
if(b==-1)
b=2;
else
b=b*10+2;
}

if(e.getSource()==b3)
{
if(b==-1)
b=3;
else
b=b*10+3;
}

if(e.getSource()==b4)
{
if(b==-1)
b=4;
else
b=b*10+4;
}

if(e.getSource()==b5)
{
if(b==-1)
b=5;
else
b=b*10+5;
}

if(e.getSource()==b6)
{
if(b==-1)
b=6;
else
b=b*10+6;
}

if(e.getSource()==b7)
{
if(b==-1)
b=7;
else
b=b*10+7;
}
```

calculator

```
if(e.getSource()==b8)
{
if(b==-1)
b=8;
else
b=b*10+8;
}

if(e.getSource()==b9)
{
if(b==-1)
b=9;
else
b=b*10+9;
}
if(e.getSource()==addition)
{
res=a+b;
}else if(e.getSource()==sub)
{
res=a-b;
}else if(e.getSource()==mul)
{
res=a*b;
}else if(e.getSource()==div)
{
res=a/b;
}
else if(e.getSource()==enter)
{
if(a==-1)
{
a=b;
b=-1;
}
}
if(a!=-1)
tf.setText(String.valueOf(a));

if(b!=-1)
tf2.setText(String.valueOf(b));


String result = String.valueOf(res);
tf3.setText(result);

if(e.getSource()==clear)
{
```

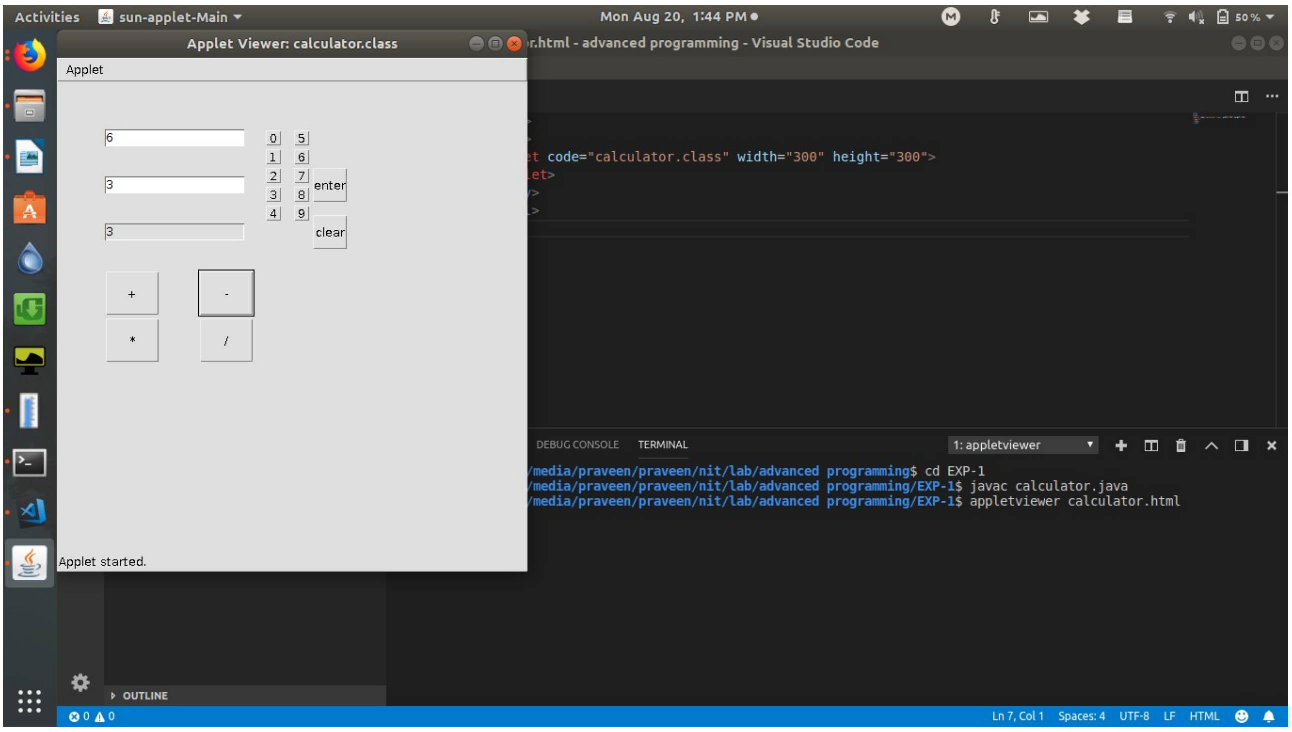
calculator

```
a=-1;
b=-1;
res=0;
tf.setText("");
tf2.setText("");
tf3.setText("");
}
}

public static void main(String[] args)
{
new calculator();
}
}
```

calculator.html file :

```
<html>
<body>
<applet code="calculator.class" width="300" height="300">
</applet>
</body>
</html>
```

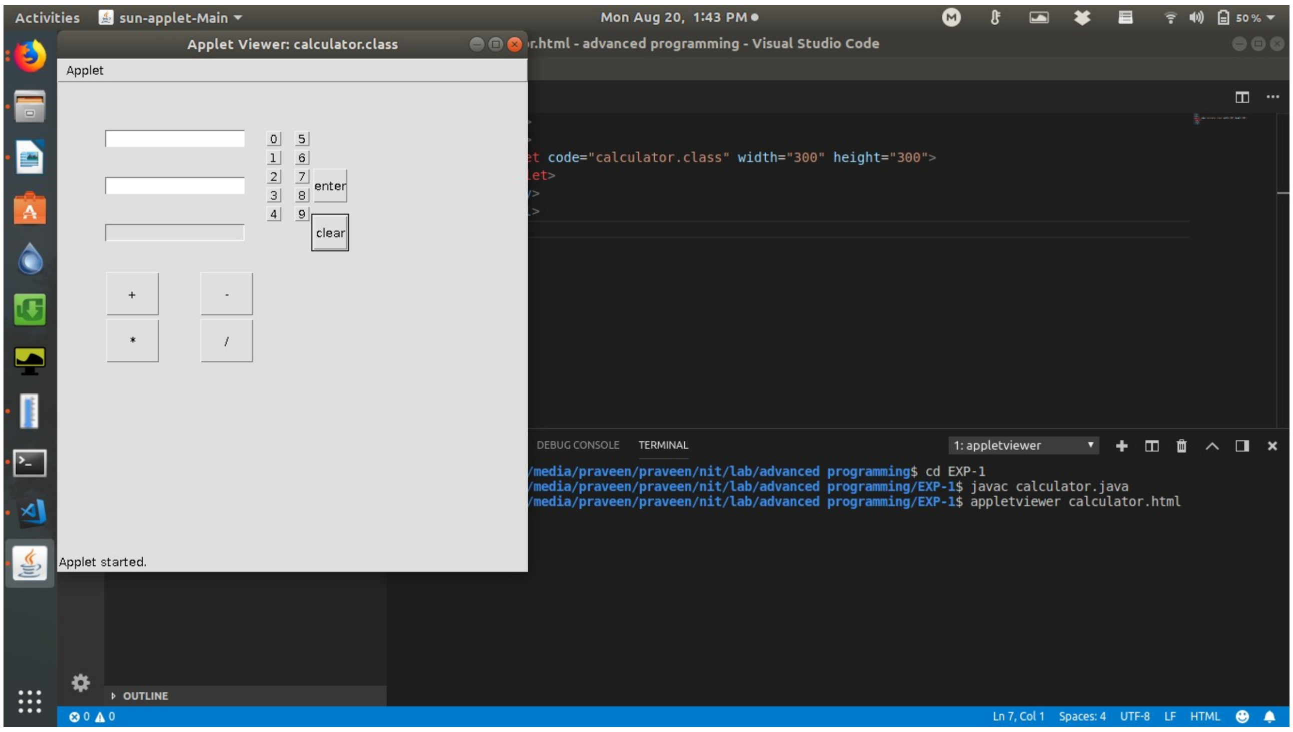
Output:



```
t code="calculator.class" width="300" height="300">
et>
/>
>
```

```
/media/praveen/praveen/nit/lab/advanced programming$ cd EXP-1
/media/praveen/praveen/nit/lab/advanced programming/EXP-1$ javac calculator.java
/media/praveen/praveen/nit/lab/advanced programming/EXP-1$ appletviewer calculator.html
```

calculator



```
et code="calculator.class" width="300" height="300">
let>
/>
l>
```

```
/media/praveen/praveen/nit/lab/advanced programming$ cd EXP-1
/media/praveen/praveen/nit/lab/advanced programming/EXP-1$ javac calculator.java
/media/praveen/praveen/nit/lab/advanced programming/EXP-1$ appletviewer calculator.html
```

car

AIM :

To draw a car using graphics options in java.

Program:

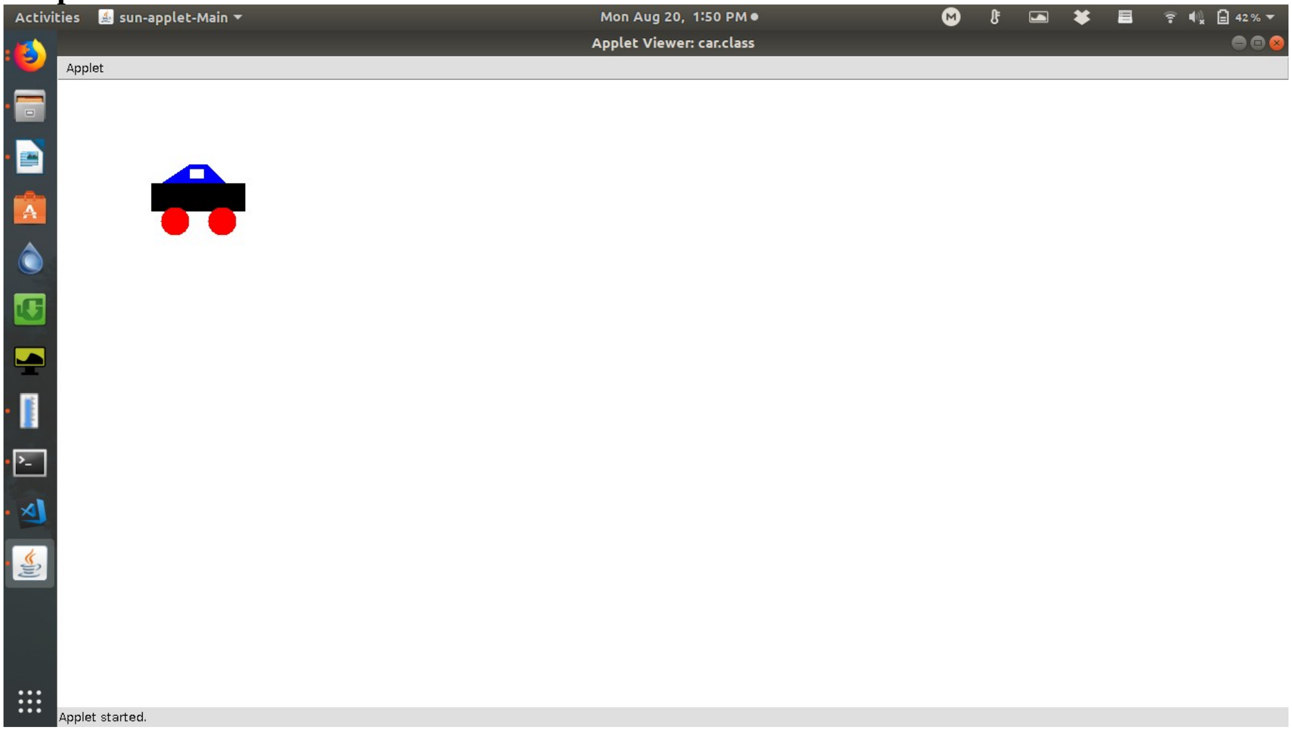
```
import java.applet.Applet;
import java.awt.Graphics;
import javax.swing.*;
import java.awt.*;

public class car extends Applet{
public void paint(Graphics g){
g.setColor(Color.white);
g.fillRect(0, 0, getWidth(), getHeight());

g.setColor(Color.black);

// drawing the car body
g.fillRect(100,110, 100, 30);

// drawing the wheels
g.setColor(Color.red);
g.fillOval(110, 135, 30, 30); // left wheel
g.fillOval(160, 135, 30, 30); // right wheel

int x[] = {110, 140, 160, 180}; // coordinate arrays for the
int y[] = {110, 90, 90, 110}; // car cabin

g.setColor(Color.blue);
g.fillPolygon(x, y, 4); // drawing the cabin in blue

g.setColor(Color.white);
g.fillRect(141,95,15,10);
}
}
```

car.html :

```
<html>
<body>
<applet code="car.class" width="300" height="300">
</applet>
</body>
</html>
```

car

Output:


Activities
sun-applet-Main
Mon Aug 20, 1:50 PM
42 %
Applet Viewer: car.class
Applet
Applet started.

Interface

AIM :

To learn the basis of the interface and inheritance and implement it for calculating area and volume for different shapes.

Program:

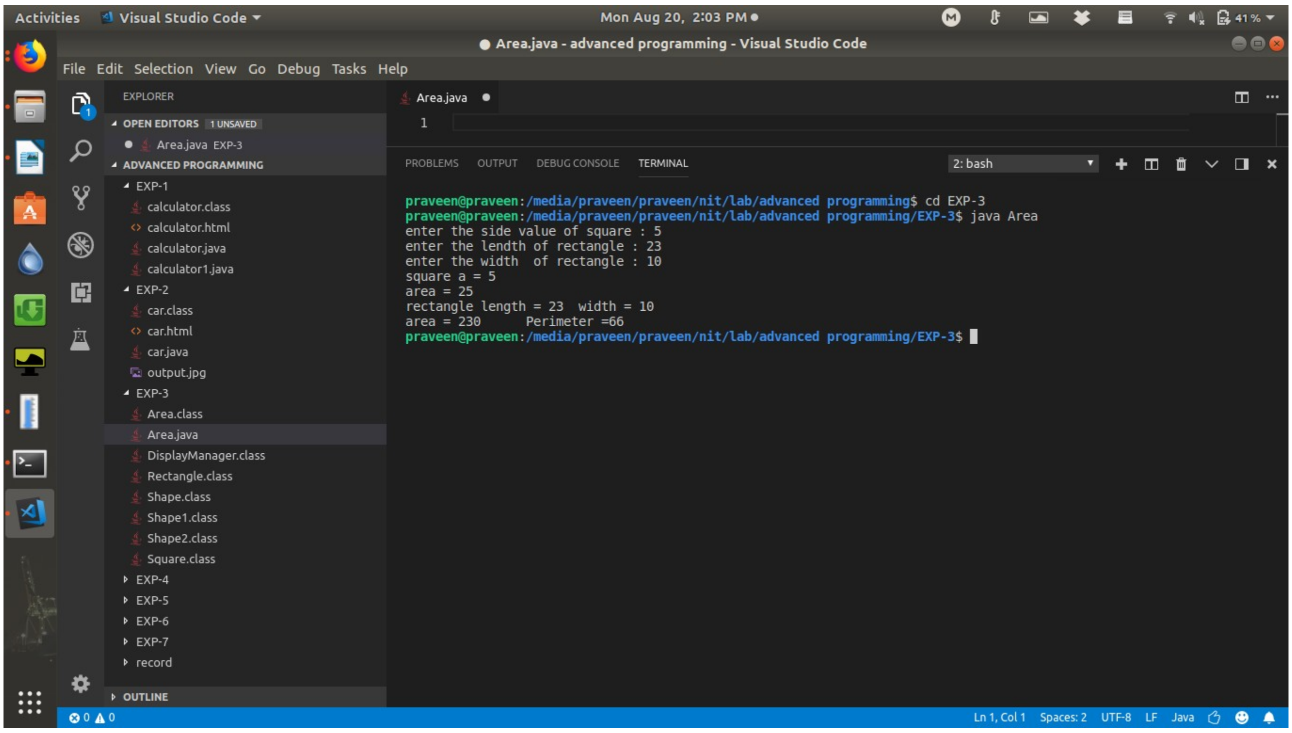
```
import java.util.Scanner;

interface Shape1
{
void Area();
}

interface Shape2
{
void Area();
void Perimeter();
}

interface DisplayManager
{
void Display();
}
class Square implements Shape1,DisplayManager
{
int a;
int area;
Square(int b)
{
a=b;
area = 0;
}
public void Area()
{
area = a*a;
}

public void Display()
{
System.out.println("area = "+area);
}
}

class Rectangle implements Shape1,Shape2,DisplayManager
{
int length,width;
int area,perimeter;
Rectangle(int l,int w)
{
length=l;
```

Interface

```
width=w;
area=0;
perimeter=0;
}

public void Area()
{
area=length*width;
}
public void Perimeter()
{
perimeter=2*(length+width);
}

public void Display()
{
System.out.println("area = "+area+"\tPerimeter ="+perimeter);
}

}
public class Area
{

public static void main(String[] args)
{
Scanner scanner = new Scanner(System.in);
int a,l,b;
System.out.print("enter the side value of square : ");
a=scanner.nextInt();
System.out.print("enter the lendth of rectangle : ");
l=scanner.nextInt();
System.out.print("enter the width of rectangle : ");
b=scanner.nextInt();
Square square = new Square(a);
square.Area();
System.out.println("square a = "+a+"\t");
square.Display();
Rectangle rectangle = new Rectangle(l,b);
rectangle.Area();
rectangle.Perimeter();
System.out.println("rectangle length = "+l+" width = "+b);rectangle.Display();

}
}
```

Interface

Output:



```
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming$ cd EXP-3
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-3$ java Area
enter the side value of square : 5
enter the lendth of rectangle : 23
enter the width  of rectangle : 10
square a = 5
area = 25
rectangle length = 23  width = 10
area = 230      Perimeter =66
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-3$
```

Multithreading

AIM :

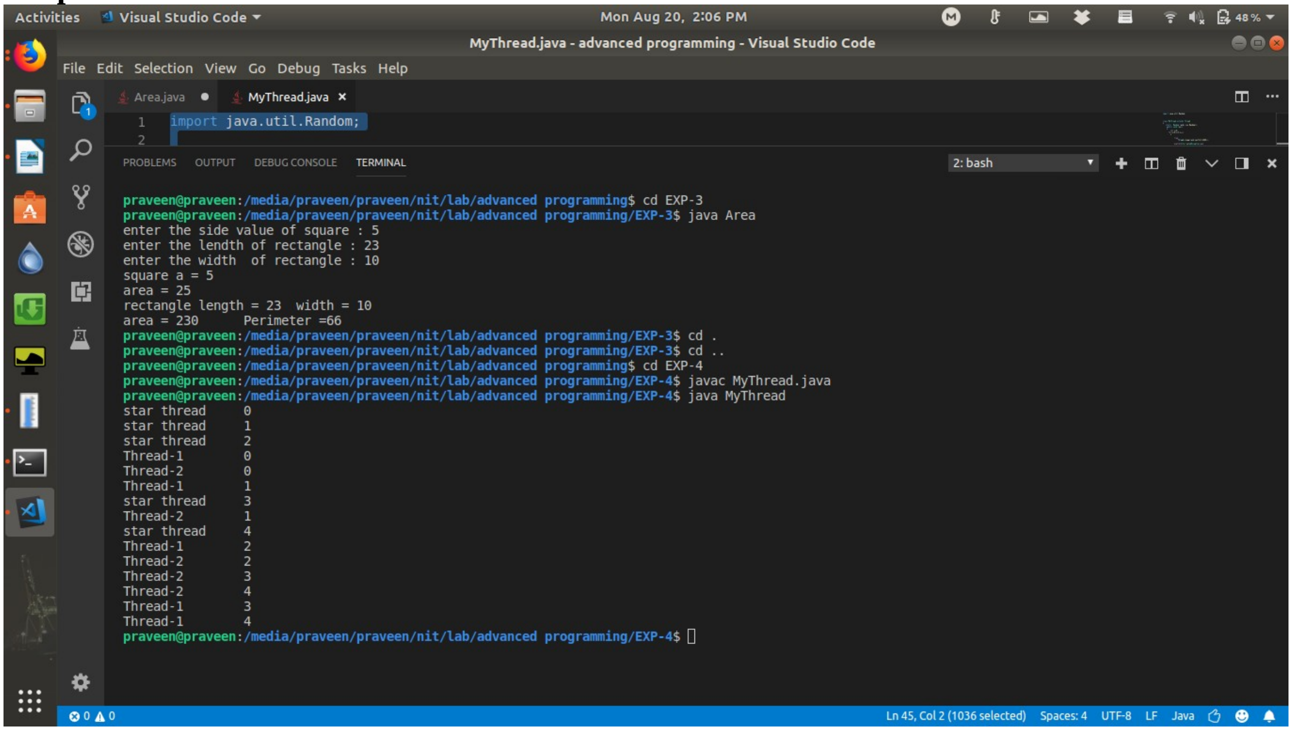
To implement basic functions of thread using java thread libraries in java.

Program:

```
import java.util.Random;


class MyThread extends Thread
{
static Random rand= new Random();
public void run()
{
int i=0;
for(i=0;i<5;i++)
{
try{
Thread.sleep(rand.nextInt(2000));

}catch(InterruptedException e){
System.out.println(e);
}
System.out.println(Thread.currentThread().getName()+"\t"+i);
}
}

public static void main(String[] args)
{
MyThread t1 = new MyThread();
t1.setName("star thread");
MyThread t2 = new MyThread();
t2.setPriority(Thread.MAX_PRIORITY);
MyThread t3 = new MyThread();
t3.setPriority(8);
t1.start();
try{
// join method waits for 3000 ms for t1 thread to finish after that only t2 and t3 will be started
t1.join(3000);
}catch(Exception e){
System.out.println(e);
}
t2.start();
t3.start();
}

}
```

Multithreading

Output:



```
import java.util.Random;
```

```
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming$ cd EXP-3
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-3$ java Area
enter the side value of square : 5
enter the lendth of rectangle : 23
enter the width  of rectangle : 10
square a = 5
area = 25
rectangle length = 23  width = 10
area = 230      Perimeter =66
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-3$ cd .
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-3$ cd ..
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming$ cd EXP-4
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-4$ javac MyThread.java
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-4$ java MyThread
star thread     0
star thread     1
star thread     2
Thread-1        0
Thread-2        0
Thread-1        1
star thread     3
Thread-2        1
star thread     4
Thread-1        2
Thread-2        2
Thread-2        3
Thread-2        4
Thread-1        3
Thread-1        4
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-4$ 
```

Method Overriding

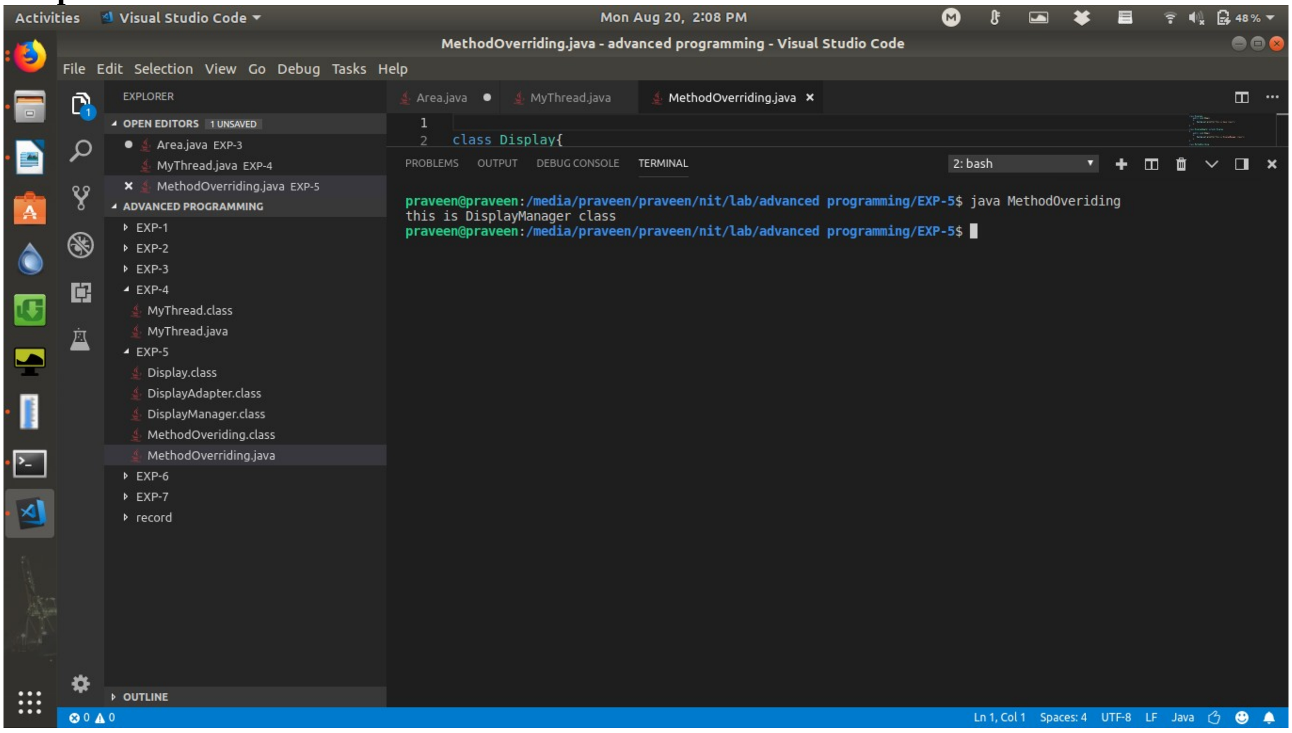
AIM :

To implement method overriding in java.

Program:

```
class Display{
public void Show()
{
System.out.println("this is base class");
}
}

class DisplayAdapter extends Display
{
public void Show()
{
System.out.println("this is DisplayManager class");
}
}

class MethodOveriding
{
public static void main(String[] args){
DisplayAdapter display = new DisplayAdapter();
display.Show();
}
}
```

Output:



```
class Display{
```

```
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-5$ java MethodOveriding
this is DisplayManager class
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-5$
```

Exceptions and Errors

AIM :

To learn about exceptions and errors and implement them in java.

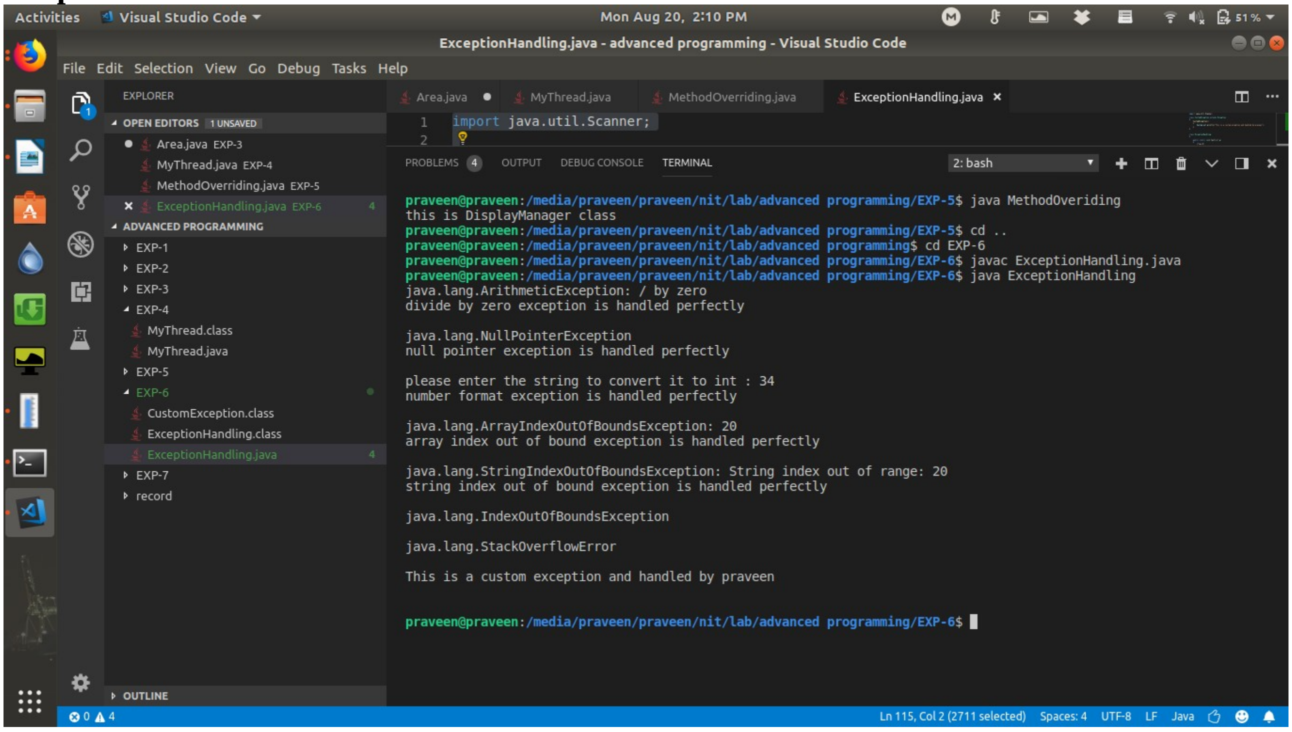
Program:

```
import java.util.Scanner;

class CustomException extends Exception
{
CustomException()
{
System.out.println("This is a custom exception and handled by praveen");
}
}

class ExceptionHandling
{
public static void test(int a)
{
if(a==5)
{
throw new IndexOutOfBoundsException();
}
}
public static void StackOverflowExceptionChecker() throws StackOverflowError
{
throw new StackOverflowError();
}
public static void CustomExceptionChecker() throws CustomException
{
throw new CustomException();
}
public static void main(String[] args)
{
Scanner scanner = new Scanner(System.in);
int data;
String str=null;
int array[]=new int[10];
String string="test";
int a=5;
try{
data=10/0;
}
catch(ArithmeticException e)
{
System.out.println(e);
}
finally
{
System.out.println("divide by zero exception is handled perfectly\n");
}

try{
str.chars();
} catch(NullPointerException e)
{
System.out.println(e);
} finally
{
System.out.println("null pointer exception is handled perfectly\n");
}

try{
System.out.print("please enter the string to convert it to int : ");
str=scanner.next();
Integer.parseInt(str);
} catch(NumberFormatException e)
{
System.out.println(e);
} finally
{
System.out.println("number format exception is handled perfectly\n");
}

try{
array[20]=20;
} catch(ArrayIndexOutOfBoundsException e)
{
System.out.println(e);
} finally
{
System.out.println("array index out of bound exception is handled perfectly\n");
}

try{
string.charAt(20);
} catch(StringIndexOutOfBoundsException e)
{
System.out.println(e);
} finally
{
System.out.println("string index out of bound exception is handled perfectly\n");
}

try{
test(a);
}catch(IndexOutOfBoundsException e)
{
System.out.println(e+"\n");
}
```

Exceptions and Errors

```
try{
StackOverflowExceptionChecker();
}catch(StackOverflowError e)
{
System.out.println(e+"\n");
}

try{
CustomExceptionChecker();
}catch(CustomException e)
{
System.out.println("\n");
}
}
}
```

Output:



```
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-5$ java MethodOveriding
this is DisplayManager class
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-5$ cd ..
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming$ cd EXP-6
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-6$ javac ExceptionHandling.java
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-6$ java ExceptionHandling
java.lang.ArithmeticException: / by zero
divide by zero exception is handled perfectly

java.lang.NullPointerException
null pointer exception is handled perfectly

please enter the string to convert it to int : 34
number format exception is handled perfectly

java.lang.ArrayIndexOutOfBoundsException: 20
array index out of bound exception is handled perfectly

java.lang.StringIndexOutOfBoundsException: String index out of range: 20
string index out of bound exception is handled perfectly

java.lang.IndexOutOfBoundsException

java.lang.StackOverflowError

This is a custom exception and handled by praveen


praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-6$ 
```

Sorting

AIM :

To implement bubble sort,insertion sort,selection sort and quick sort in java.

Program:

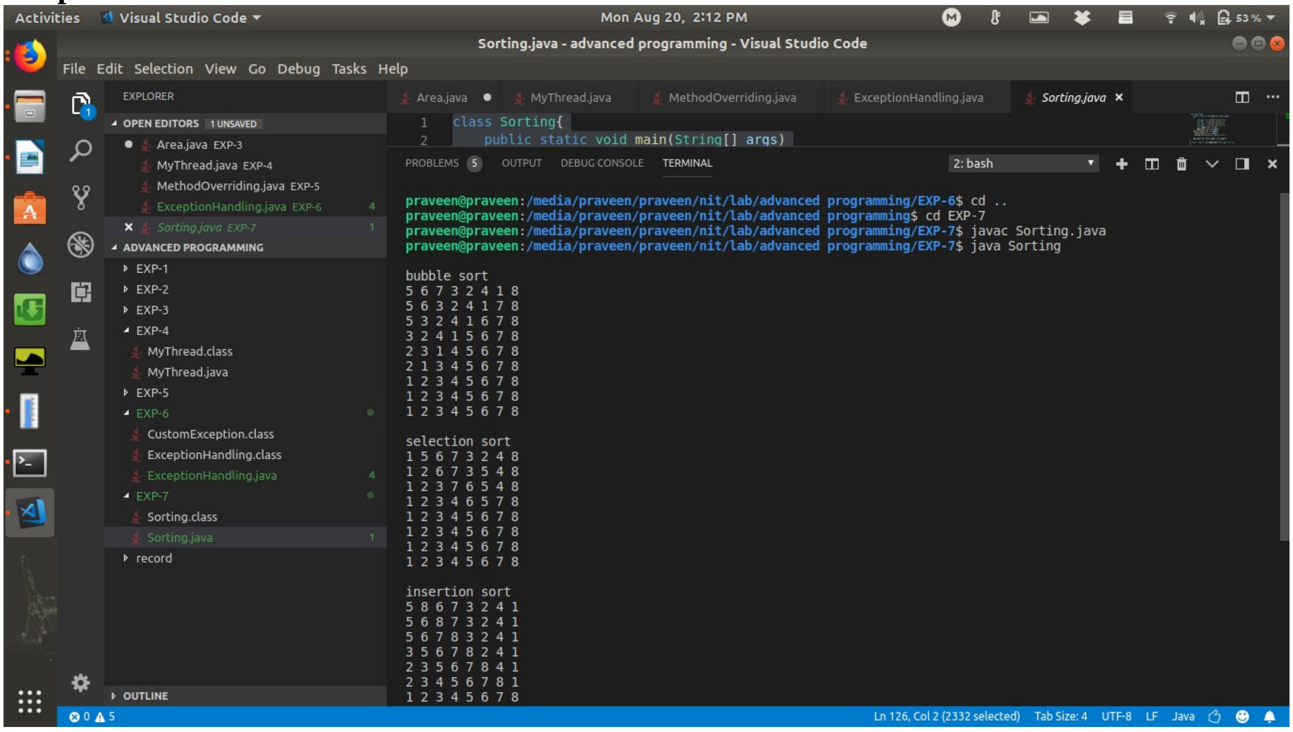
```
class Sorting{
public static void main(String[] args)
{
int a[] = { 8,5,6,7,3,2,4,1};
     int b[] = { 8,5,6,7,3,2,4,1 };
     int c[] = { 8,5,6,7,3,2,4,1};
     int d[] = { 8,5,6,7,3,2,4,1 };
int temp[]=new int[8];
BubbleSort(a, a.length);
    SelectionSort(b, b.length);
     InsertionSort(c,c.length);
System.out.println("\nquick sort");

QuickSort(d, 0,d.length-1,d.length);
}
private static void BubbleSort(int a[], int n)
{
System.out.println("\nbubble sort");
    int pass, i,temp, swapped = 1;
    for (pass = n - 1; pass >= 0 && swapped==1; pass--)
    {
        swapped = 0;
        for (i = 0; i < pass; i++)
        {
            if (a[i] > a[i + 1])
            {
                temp = a[i];
                a[i] = a[i + 1];
                a[i + 1] = temp;
                swapped = 1;
            }
        }
Display(a, n);
    }
    Display(a,n);
}
private static void SelectionSort(int a[], int n)
{
System.out.println("\nselection sort");

    int i, j, min, temp;
    for (i = 0; i < n - 1; i++)
    {
        min = i;
        for (j = i + 1; j < n; j++)
```

Sorting

```
        {
            if (a[j] < a[min])
                min = j;
        }
        temp = a[min];
        a[min] = a[i];
        a[i] = temp;
Display(a,n);
    }
    Display(a, n);
}

private static void InsertionSort(int a[], int n)
{
System.out.println("\ninsertion sort");

    int i, j, value;
    for (i = 1; i < n ; i++)
    {
        j = i;
        value = a[j];
        while((j>=1)&&(a[j-1] > value))
        {
            a[j] = a[j - 1];
            j--;
        }

        a[j] = value;
Display(a, n);
}
    Display(a, n);
}

public static void QuickSort(int A[], int low, int high,int n)
{
    int pivot;
    if (high > low)
    {
        pivot = Partition(A, low, high);
System.out.println("pivot : "+A[pivot]);
Display(A,n);
QuickSort(A, low, pivot - 1,n);
        QuickSort(A, pivot + 1, high,n);
    }
//  if (low == 0 && high == n-1)
        //Display(A, n);
}

public static int Partition(int A[], int low, int high)
{
    int left, right, pivot,temp;
    left = low;
    right = high;
    pivot = A[low];
    while (left <= right)
    {
        if (left <= high && A[left] <= pivot)
            left++;
        if (right >= low && A[right] >= pivot)
            right--;
        if (left < right)
        {
            temp = A[left];
            A[left] = A[right];
            A[right] = temp;
        }
    }
    A[low] = A[right];
    A[right] = pivot;
return right;
}

private static void Display(int a[],int length)
{
int i;
for(i=0;i<length;i++)
System.out.print(a[i]+" ");
System.out.print("\n");
}

}
```

Sorting

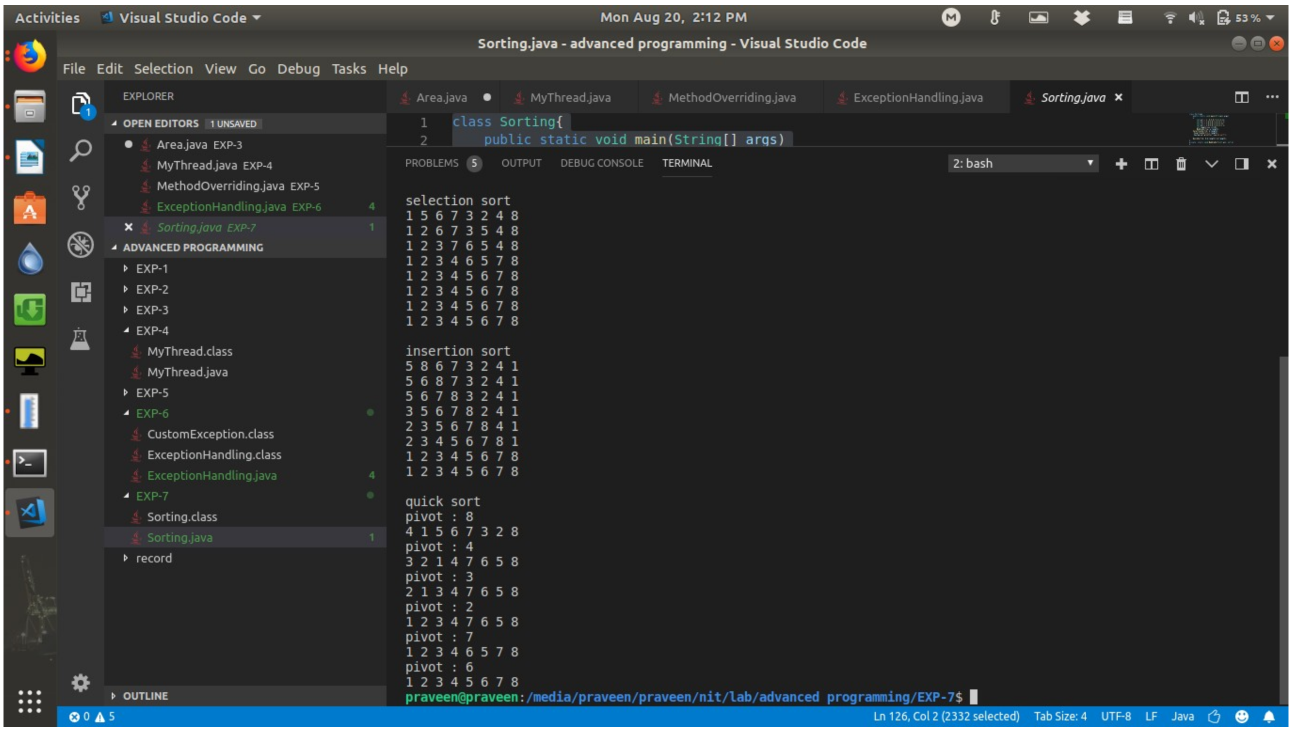
Output:



```
class Sorting{
    public static void main(String[] args)
```

```
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-6$ cd ..
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming$ cd EXP-7
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-7$ javac Sorting.java
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-7$ java Sorting

bubble sort
5 6 7 3 2 4 1 8
5 6 3 2 4 1 7 8
5 3 2 4 1 6 7 8
3 2 4 1 5 6 7 8
2 3 1 4 5 6 7 8
2 1 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8

selection sort
1 5 6 7 3 2 4 8
1 2 6 7 3 5 4 8
1 2 3 7 6 5 4 8
1 2 3 4 6 5 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8

insertion sort
5 8 6 7 3 2 4 1
5 6 8 7 3 2 4 1
5 6 7 8 3 2 4 1
3 5 6 7 8 2 4 1
2 3 5 6 7 8 4 1
2 3 4 5 6 7 8 1
1 2 3 4 5 6 7 8
```

Sorting



```
class Sorting{
    public static void main(String[] args)
```

```
selection sort
1 5 6 7 3 2 4 8
1 2 6 7 3 5 4 8
1 2 3 7 6 5 4 8
1 2 3 4 6 5 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8

insertion sort
5 8 6 7 3 2 4 1
5 6 8 7 3 2 4 1
5 6 7 8 3 2 4 1
3 5 6 7 8 2 4 1
2 3 5 6 7 8 4 1
2 3 4 5 6 7 8 1
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8

quick sort
pivot : 8
4 1 5 6 7 3 2 8
pivot : 4
3 2 1 4 7 6 5 8
pivot : 3
2 1 3 4 7 6 5 8
pivot : 2
1 2 3 4 7 6 5 8
pivot : 7
1 2 3 4 6 5 7 8
pivot : 6
1 2 3 4 5 6 7 8
praveen@praveen:/media/praveen/praveen/nit/lab/advanced programming/EXP-7$
```