Topics Covered

- Arrays
- Inheritance
- Polymorphism
- Exception Handling

Array

- What is array?
  - Java provides a data structure, the **Array**, which stores a fixed-size sequential collection of elements of the same type.
  - A specific element in array is accessed by its **index**.
- Example :
  - Instead of declaring individual variables, for example, num0, num1, num2…., we can declare one array variable, “num” and num[0], num[1], num[2] is used to access individual variables.

Array

- Array be of one or multiple dimension, first we will learn “one dimensional array”.
- *Declaring array variables*
  - General form of an one dimensional array declaration is:
    - `Type var-name[ ];`
  - Type declares **base type**(the type of which all elements in the array is) of an array
  - Example : `int our_first_array[];`

Array

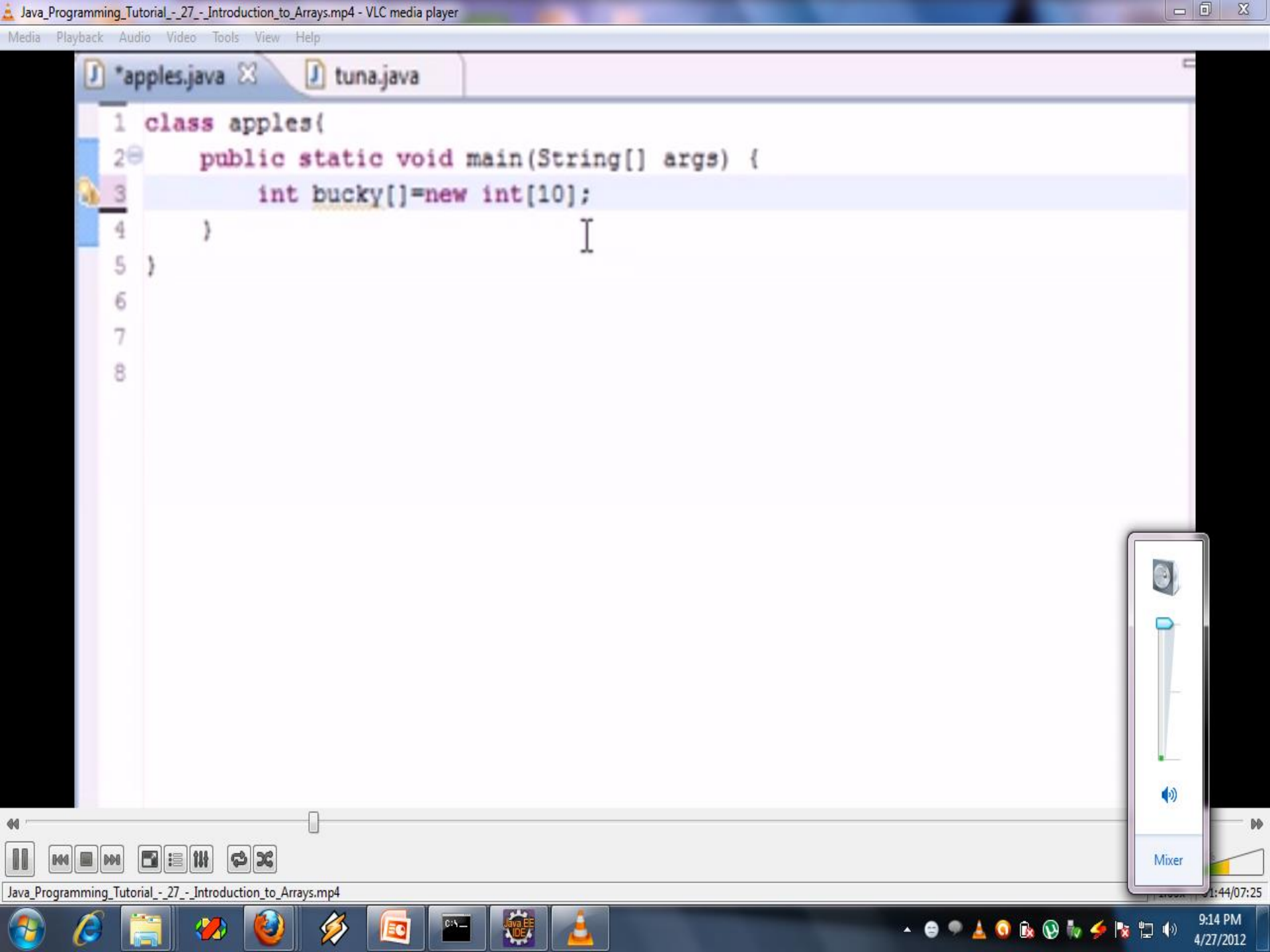
- This declaration establishes that *our_first_array* is an array variable, but it is not created yet.
- Our_first_array returns null if it is tried to be accessed, because it is not given any value.

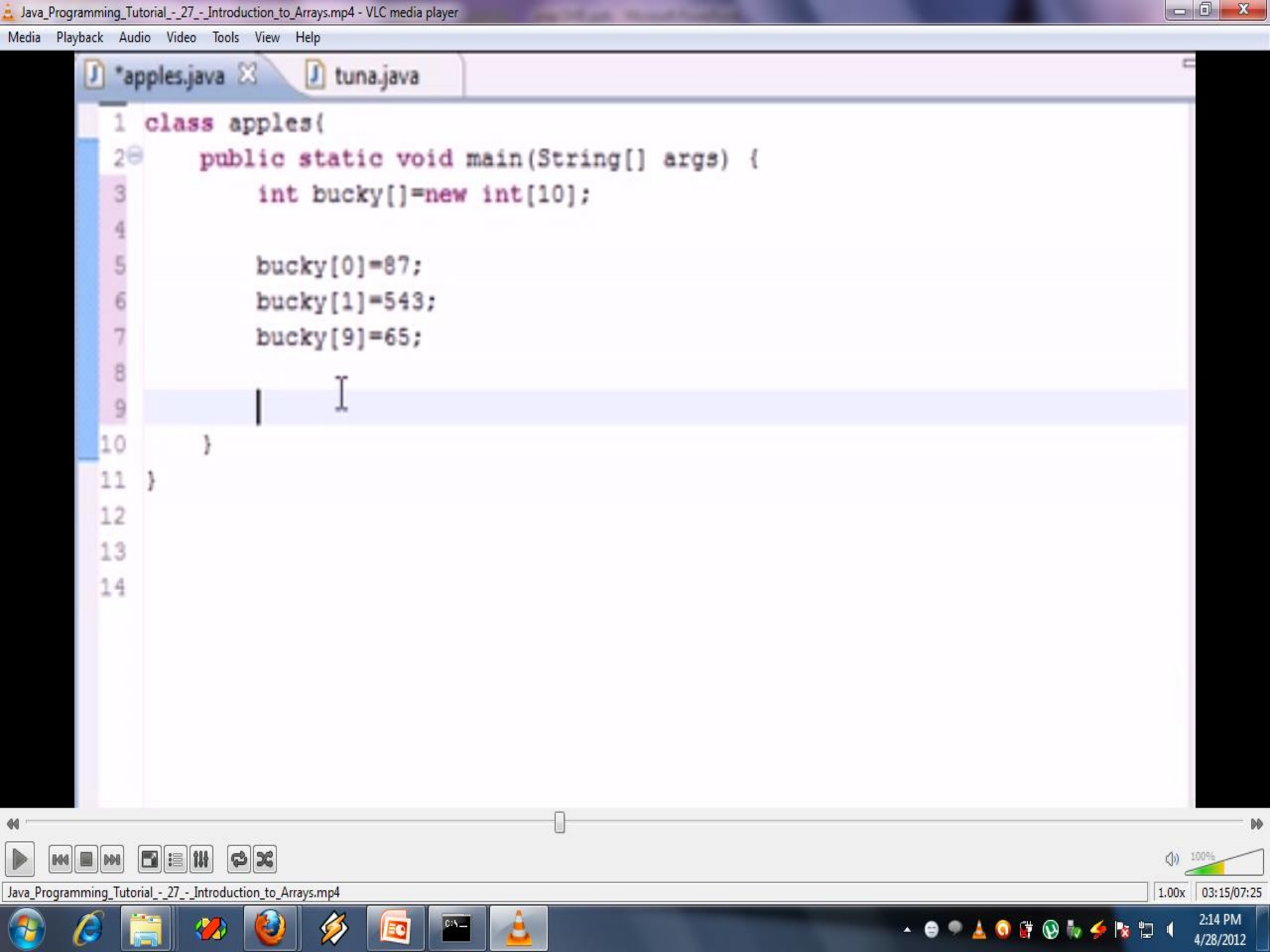
Array

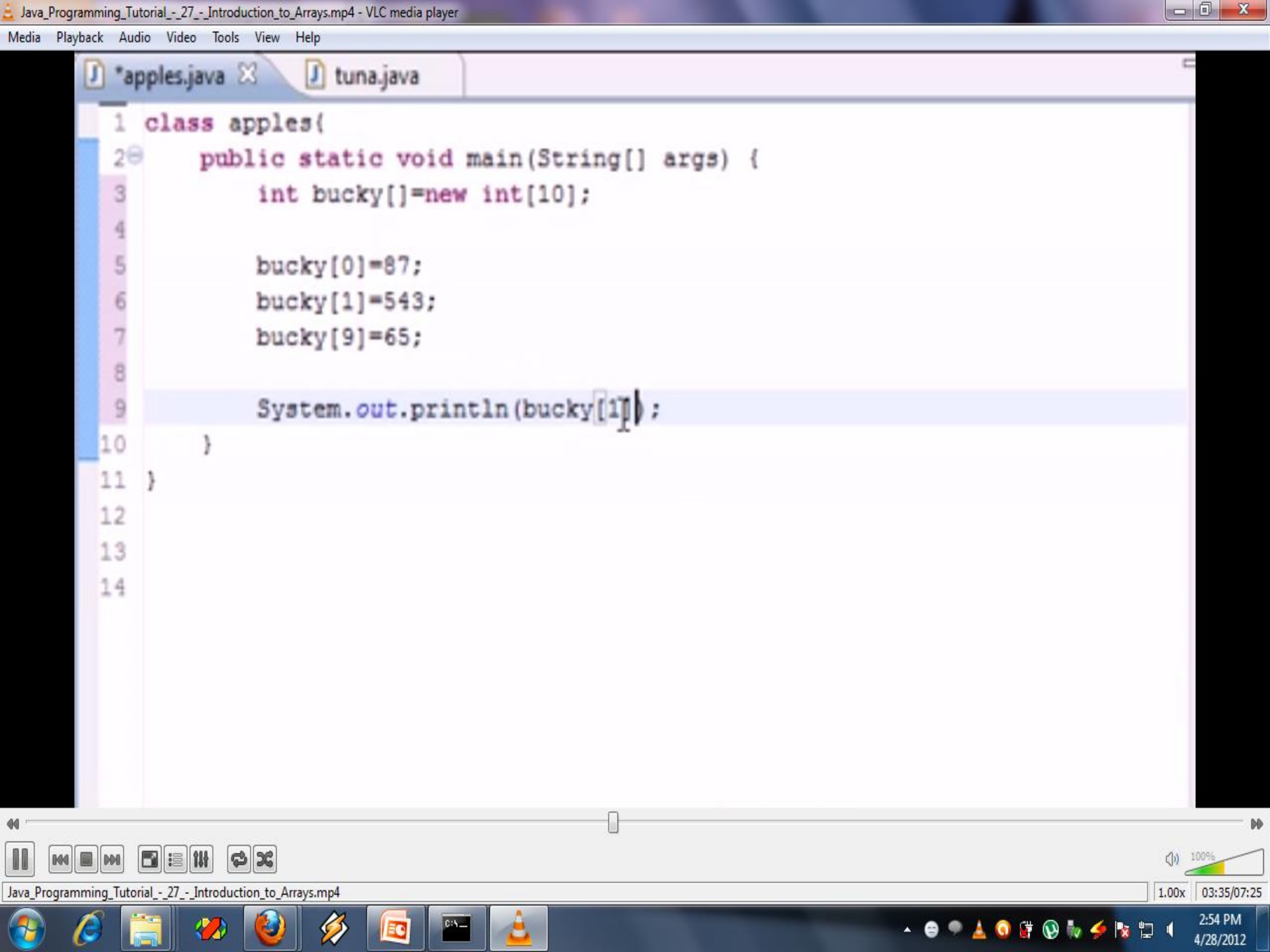
- *Creating Arrays* :
  - So, to actually create array variable, we have to allocate memory for the variable, so as for each element of the array.
  - For this a operator is used, named **new**.
  - So, the general form of creating array :
    - type array-var[ ] = new type[size];
  - Here, the **size** is the number of elements of same type in the array.

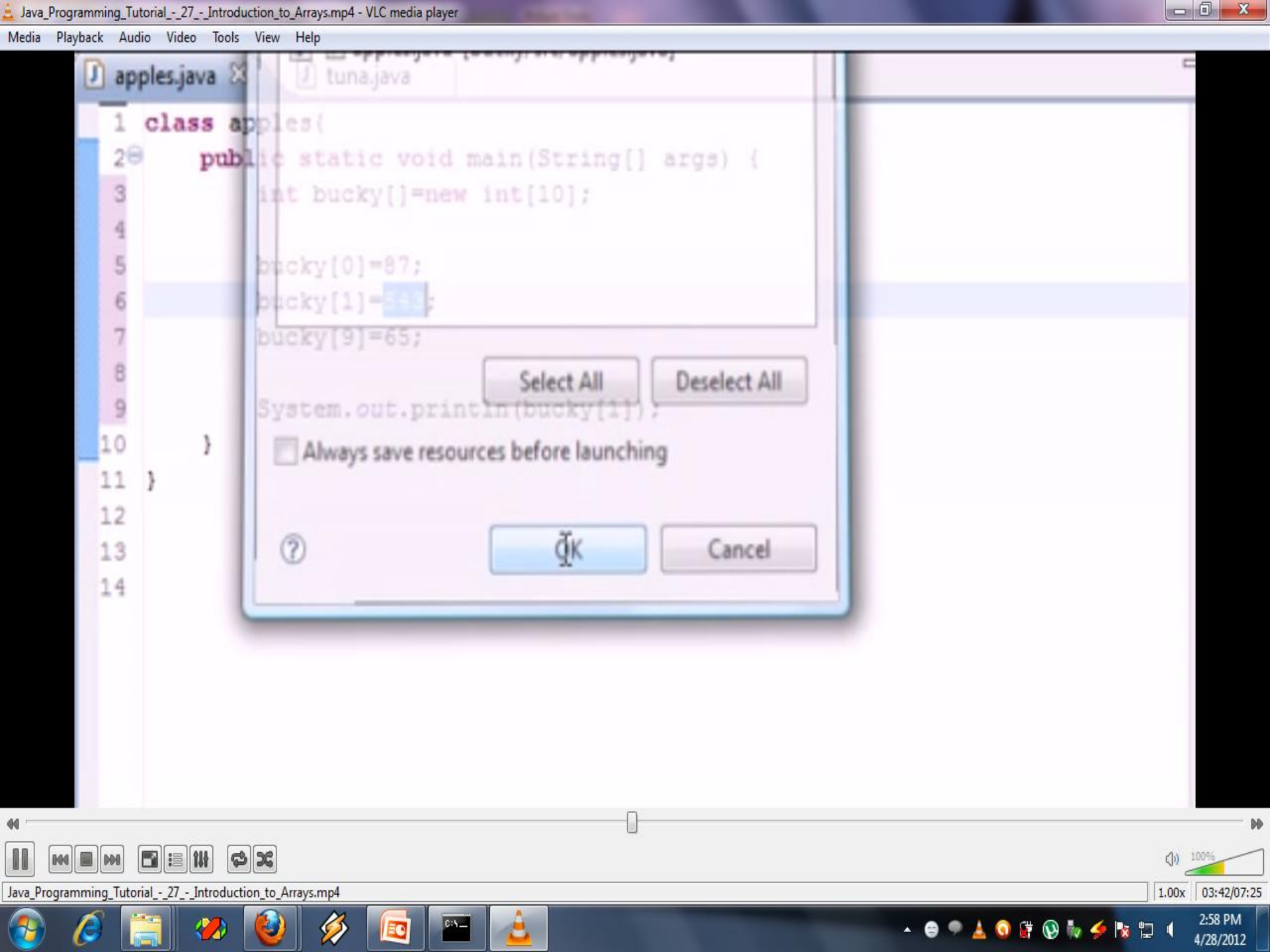
Array

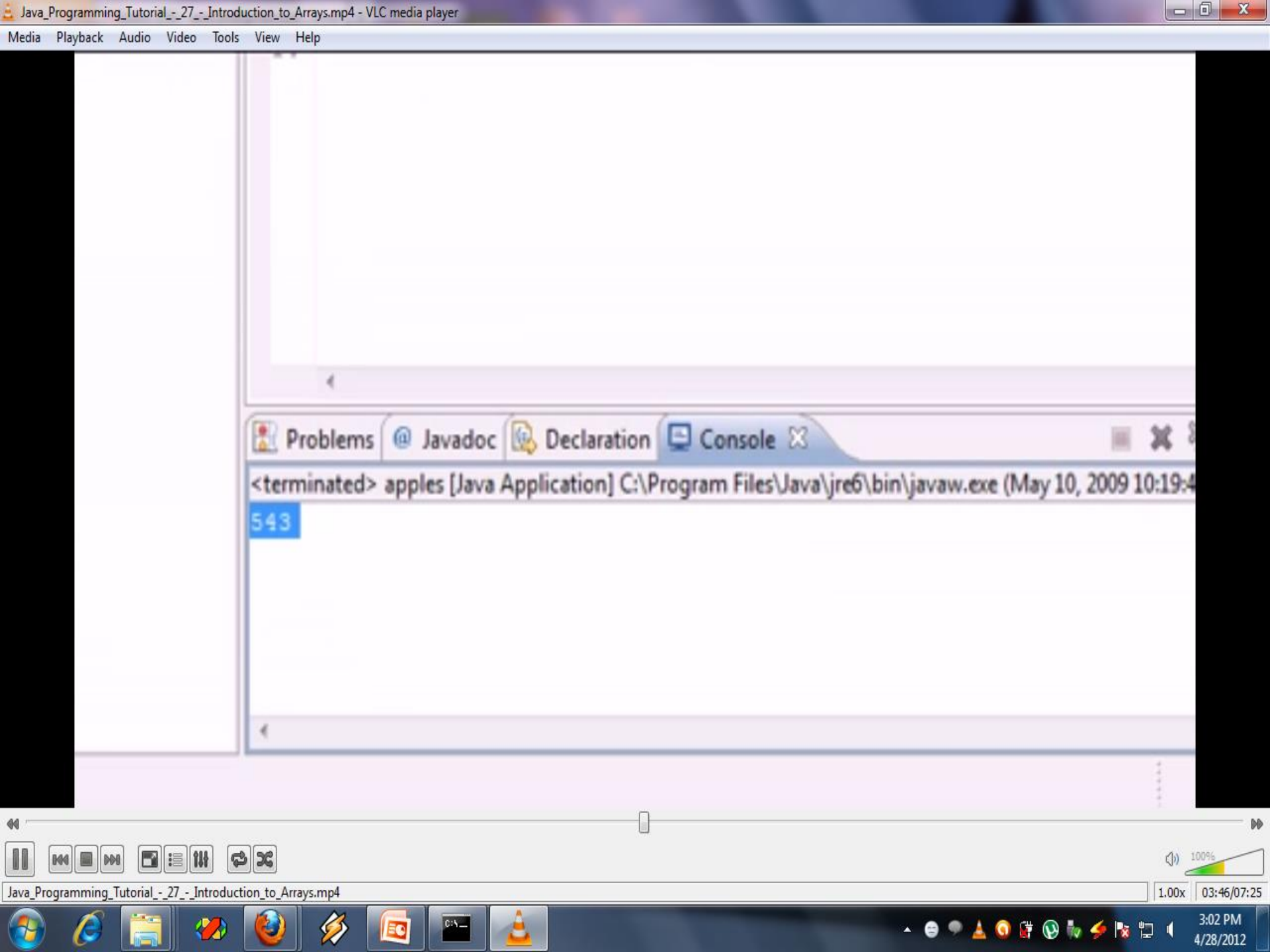
- "new" allocates memory for size amount of elements of the corresponding type.
- Example :
  - `int our_first_array[] = new int[10];`
- So the array "our_first_array" is a collection of 10 integer variable.
- And "new" allocates 10 integer size memory for 10 integer values,
- Another way :
  - `int[] our_first_array = new int[10];`


Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3 int bucky[]=new int[10];
4 }
5 }
6
7
8
Mixer
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
9:14 PM
4/27/2012


Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
class apples{
public static void main(String[] args) {
int bucky[]=new int[10];
bucky[0]=87;
bucky[1]=543;
bucky[9]=65;
}
}
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
03:15/07:25
2:14 PM
4/28/2012

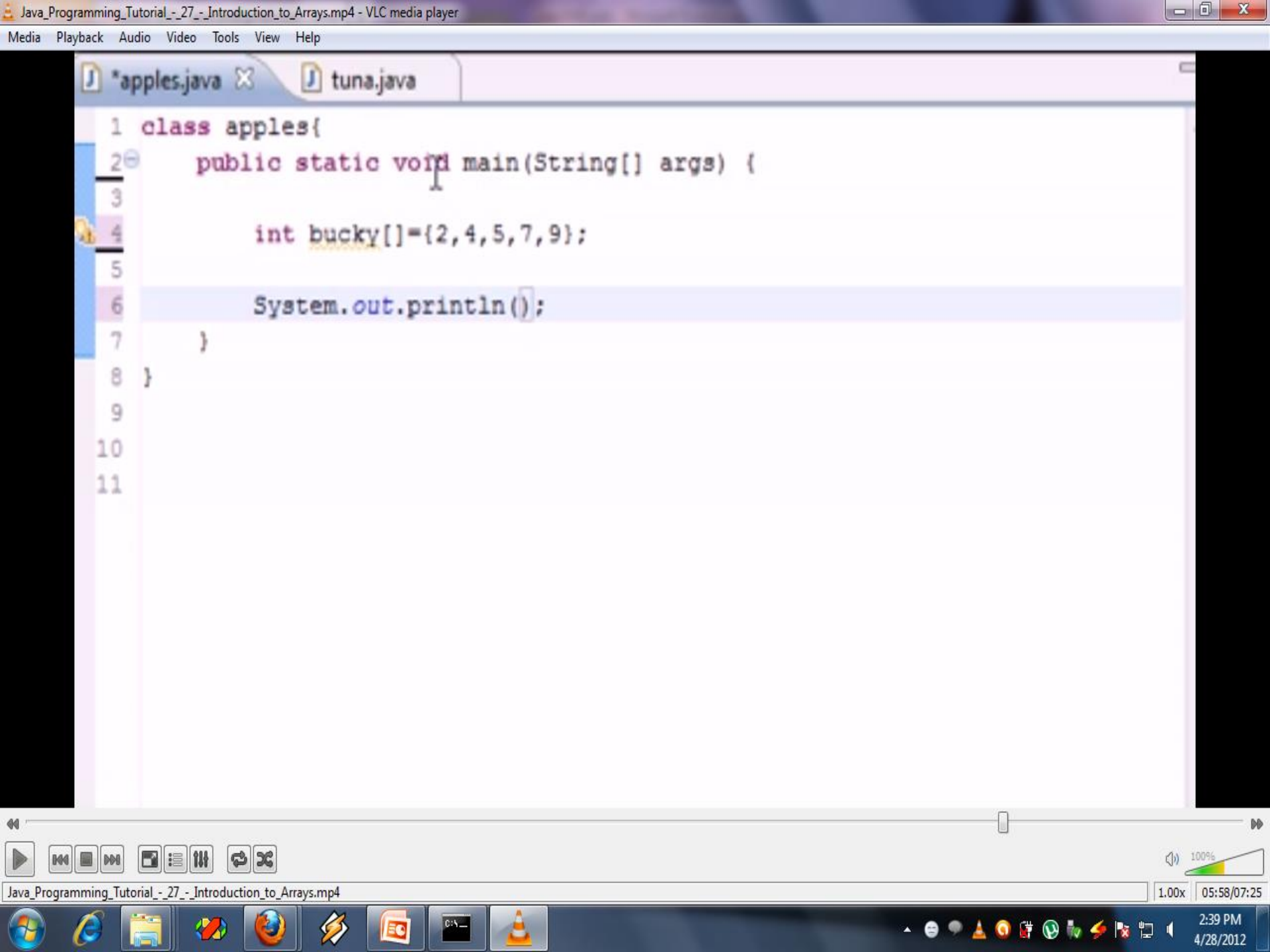

Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
class apples{
public static void main(String[] args) {
int bucky[]=new int[10];
bucky[0]=87;
bucky[1]=543;
bucky[9]=65;
System.out.println(bucky[1]);
}
}
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
03:35/07:25
2:54 PM
4/28/2012

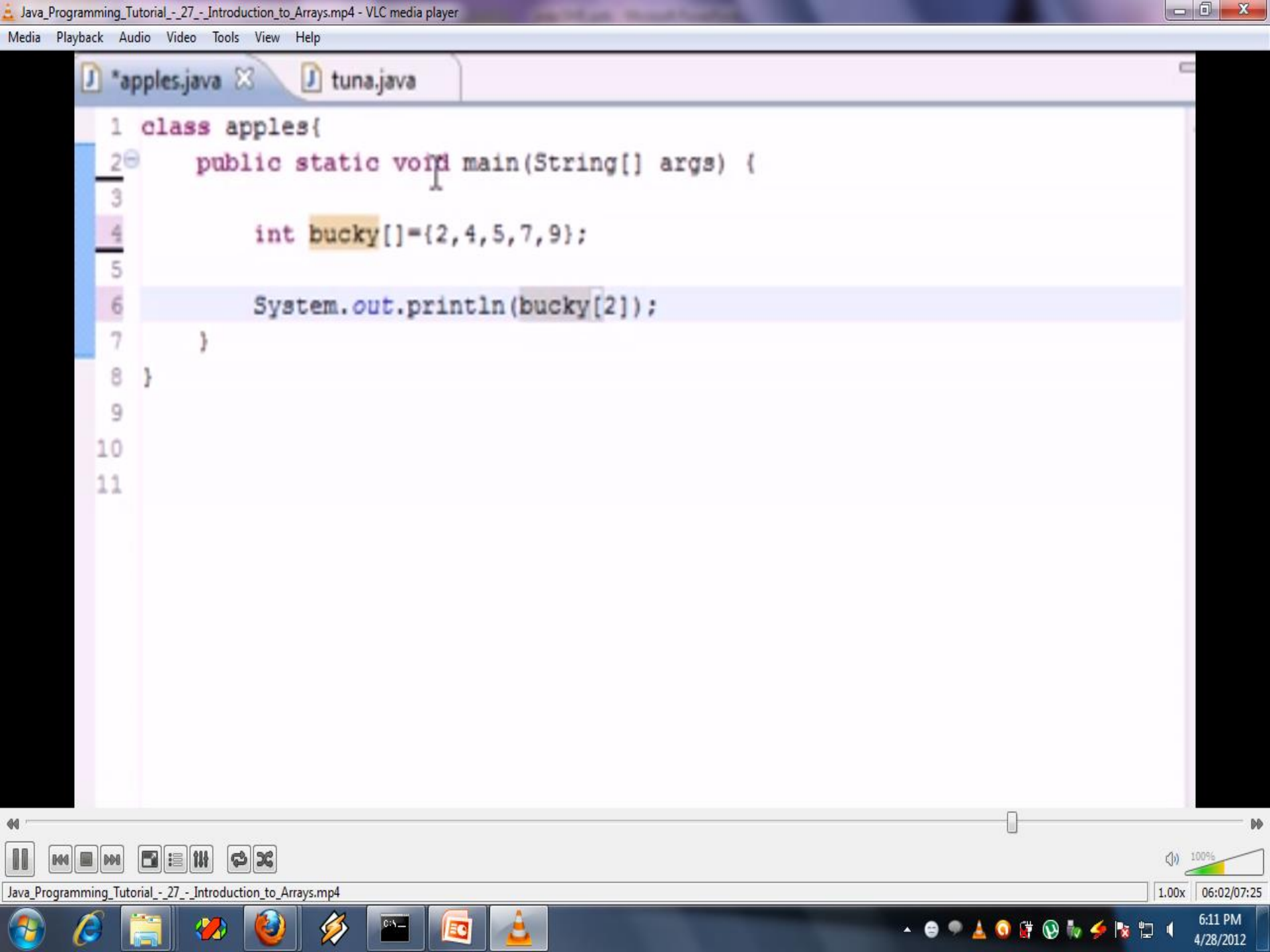

Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
apples.java
tuna.java
class apples{
public static void main(String[] args) {
int bucky[]=new int[10];
bucky[0]=87;
bucky[1]=
bucky[9]=65;
System.out.println(bucky[1]);
}
}
Select All
Deselect All
Always save resources before launching
OK
Cancel
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
03:42/07:25
2:58 PM
4/28/2012

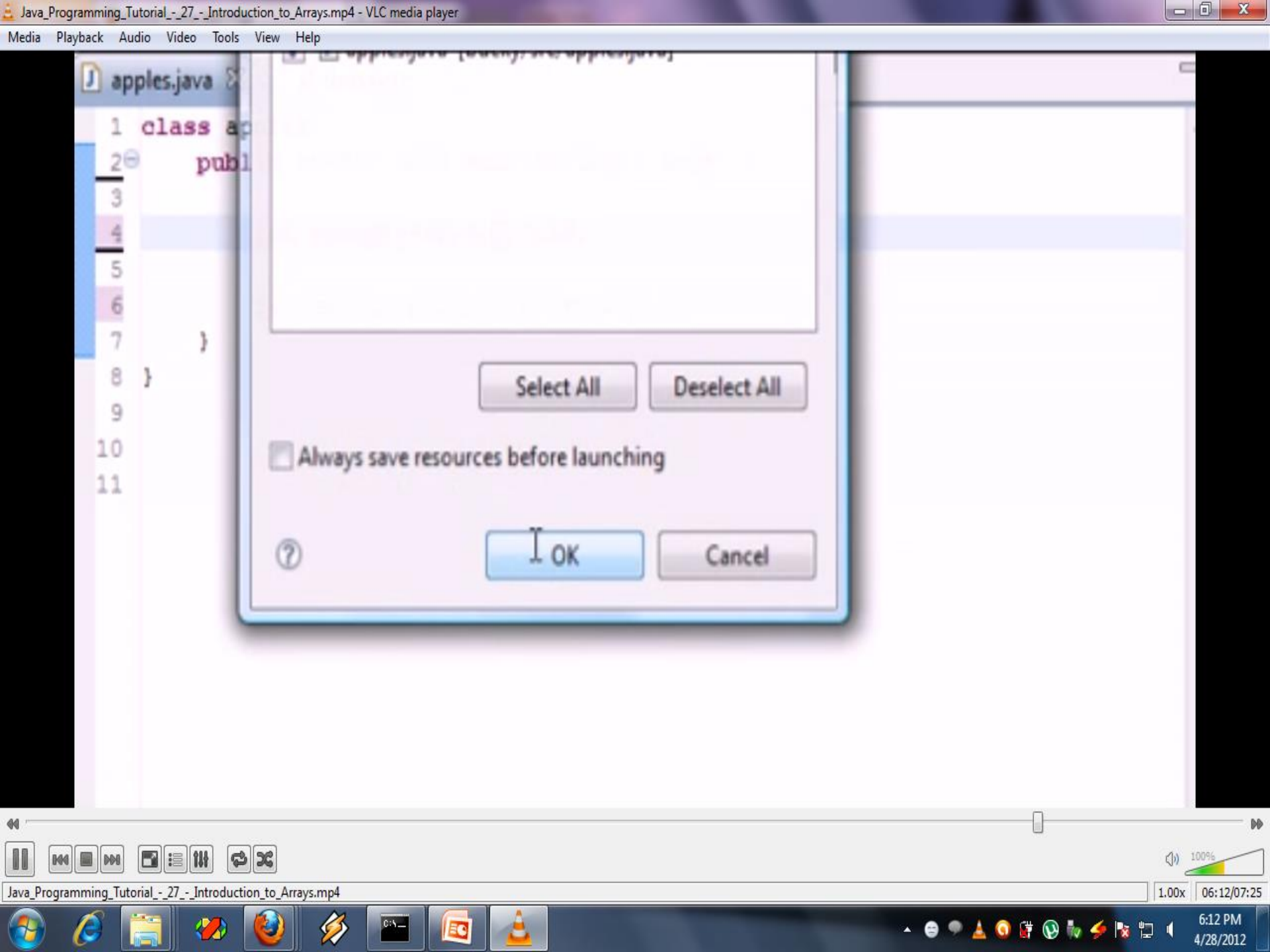

Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
Problems Javadoc Declaration Console
<terminated> apples [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 10, 2009 10:19:4
543
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
03:46/07:25
3:02 PM
4/28/2012

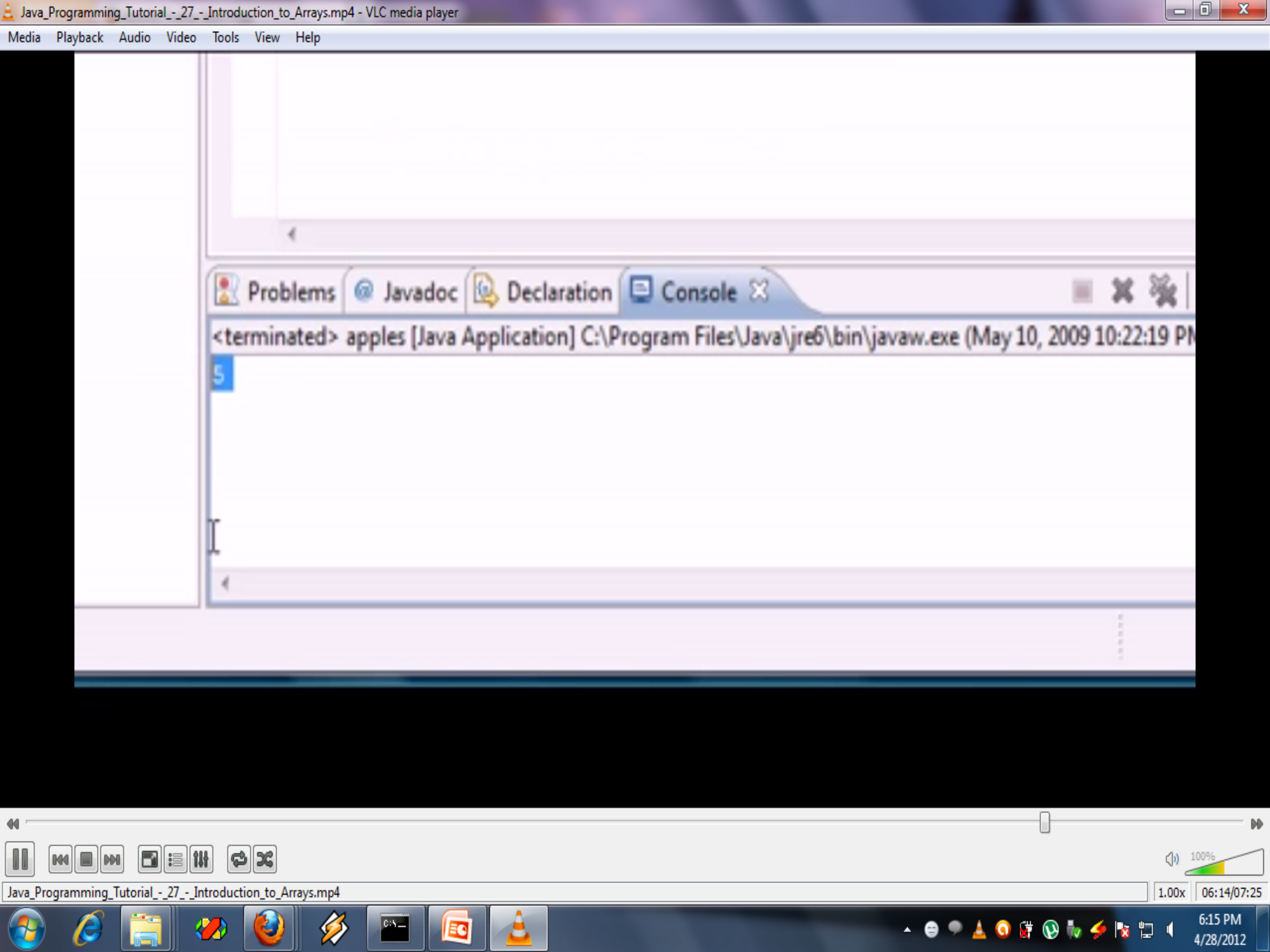
Array

- *Another way of creating and assigning values to an array :*
- The syntax will be :
  - Type array-name[] = {value1, value2, value3};
  - Here, at the same time, array declaration + creation + value assignment is done.
  - The total number of values indicates the size of the array.
  - int our_first_array[] = {10, 20, 30, 40}
  - IMP : In this process, “new” is not needed.


Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3
4 int bucky[]={2,4,5,7,9};
5
6 System.out.println();
7 }
8 }
9
10
11
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
05:58/07:25
2:39 PM
4/28/2012


Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3
4 int bucky[]={2,4,5,7,9};
5
6 System.out.println(bucky[2]);
7 }
8 }
9
10
11
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
06:02/07:25
6:11 PM
4/28/2012

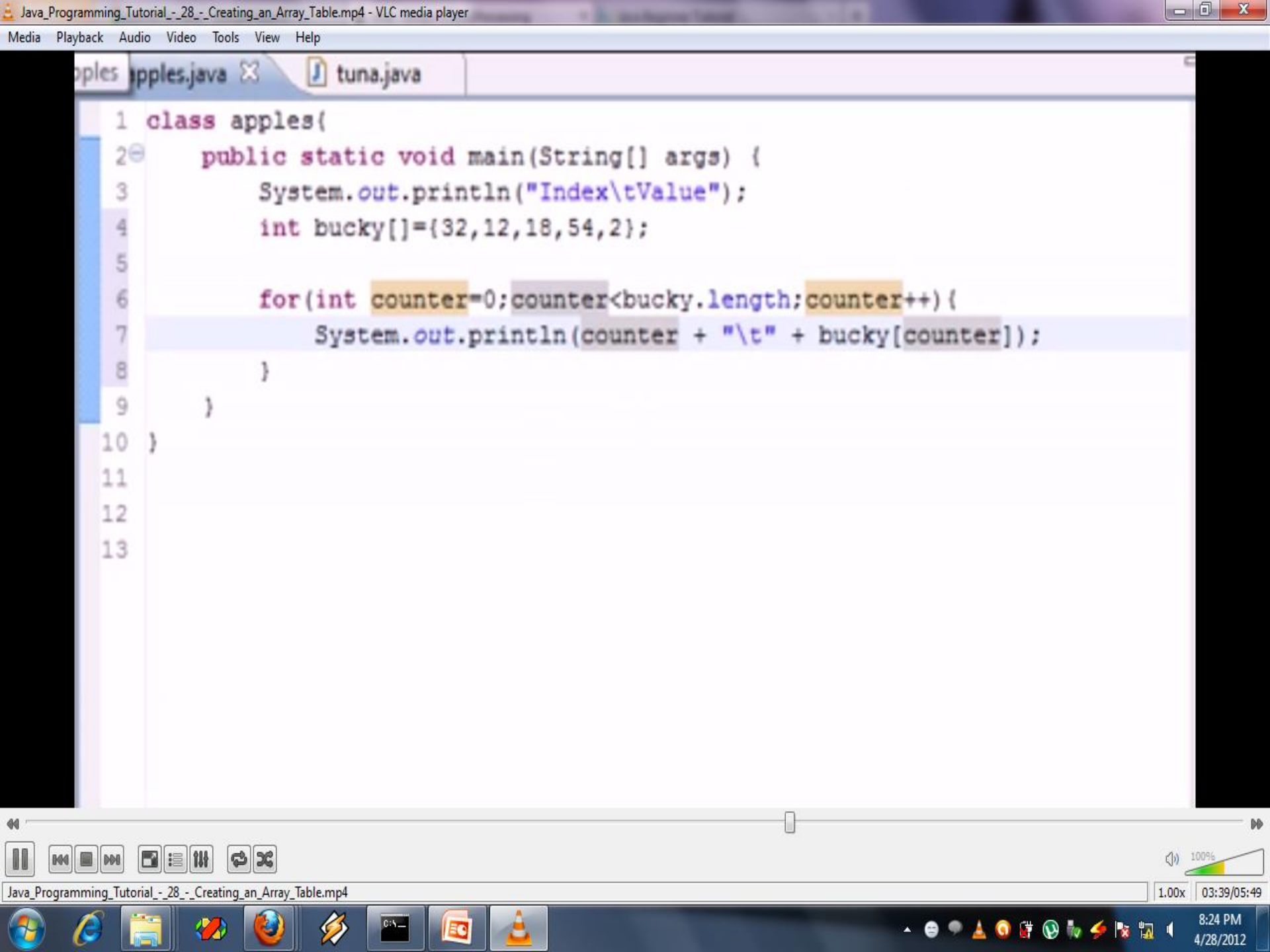

Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
apples.java
1 class a
2 publ
3
4
5
6
7 }
8 }
9
10
11
Select All
Deselect All
Always save resources before launching
OK
Cancel
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
06:12/07:25
6:12 PM
4/28/2012

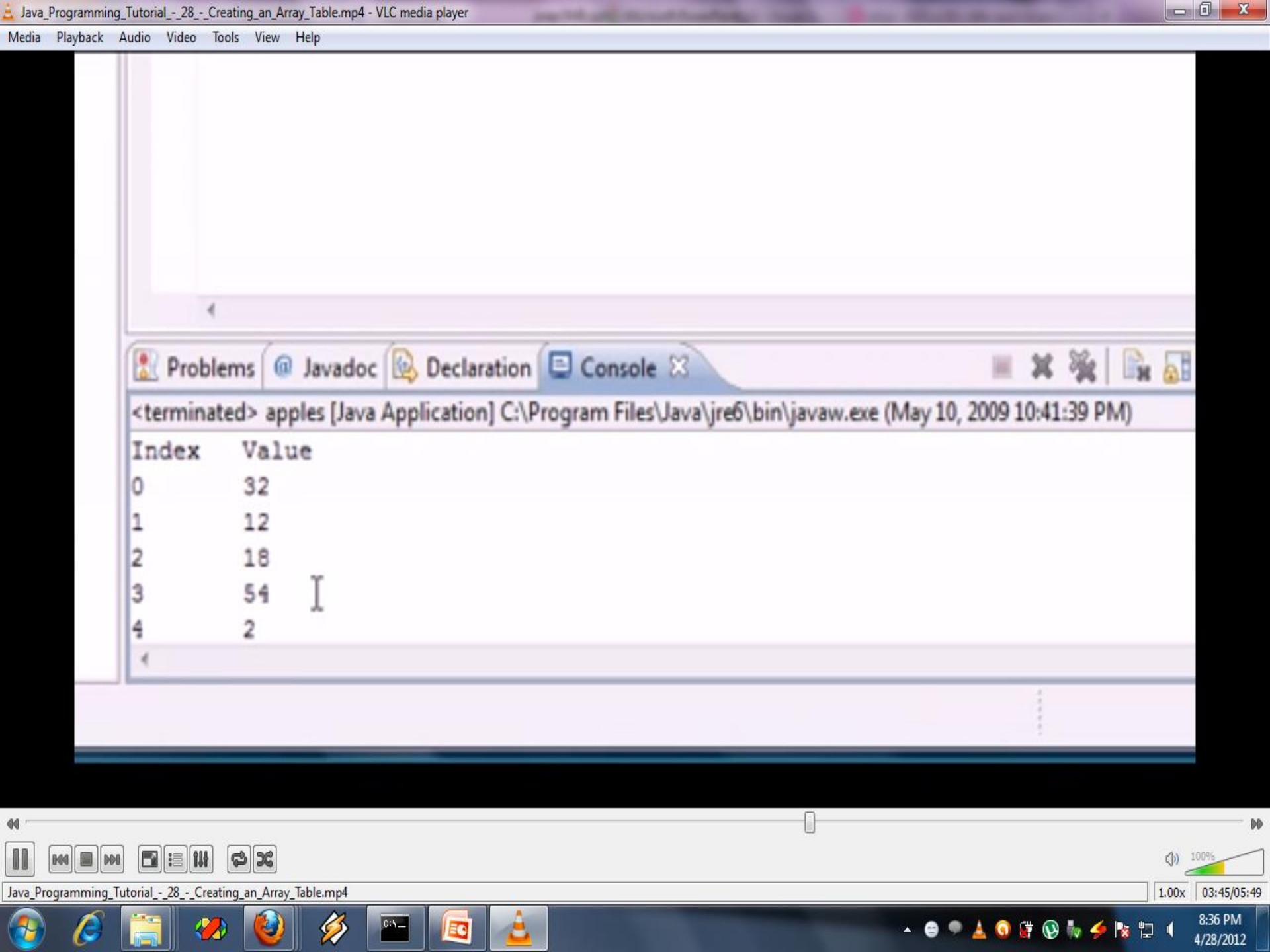

Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
Problems Javadoc Declaration Console
<terminated> apples [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 10, 2009 10:22:19 PM
5
Java_Programming_Tutorial_-_27_-_Introduction_to_Arrays.mp4
1.00x
06:14/07:25
6:15 PM
4/28/2012

Array

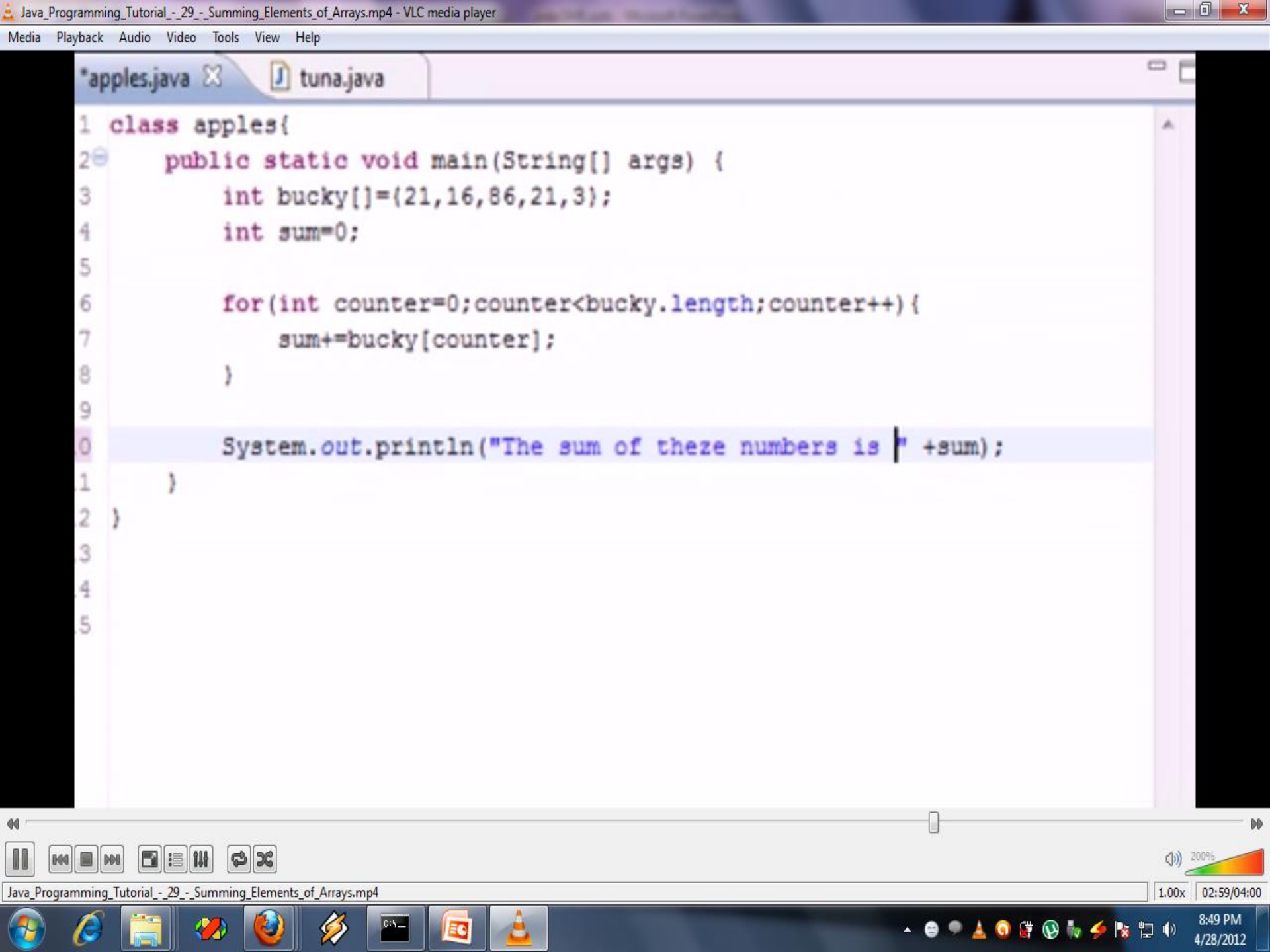
- An way of having the size of an array, for example, our_first_array is
  - array_name.**length**
  - Example : *our_first_array.length*
- Following are some examples of using array in different cases:
  - Sum up the elements of an array

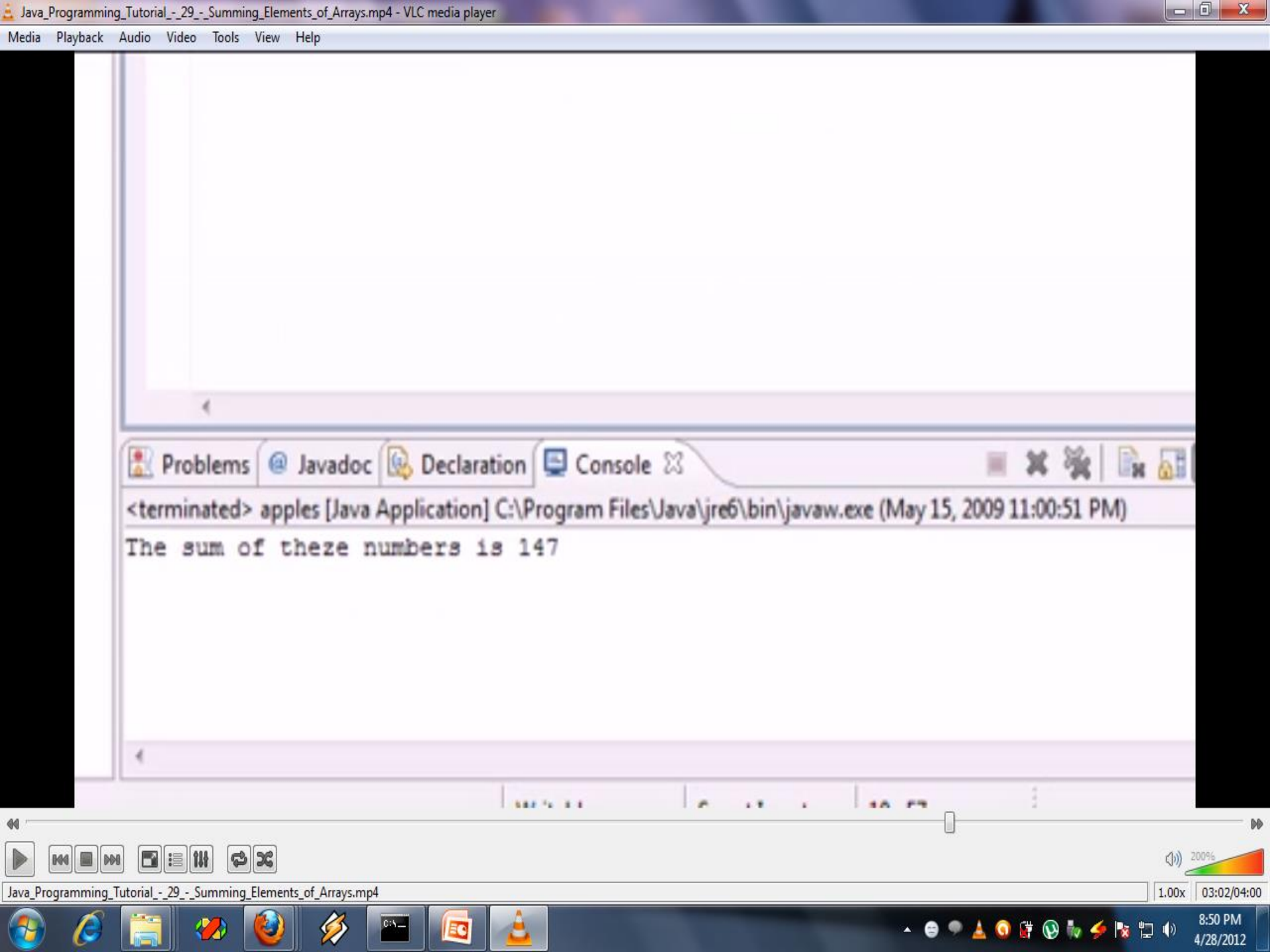
Creating table of elements of an array


Java_Programming_Tutorial_-_28_-_Creating_an_Array_Table.mp4 - VLC media player
Media Playback Audio Video Tools View Help
apples.java
tuna.java
class apples{
public static void main(String[] args) {
System.out.println("Index\tValue");
int bucky[]={32,12,18,54,2};
for(int counter=0;counter<bucky.length;counter++){
System.out.println(counter + "\t" + bucky[counter]);
}
}
}
Java_Programming_Tutorial_-_28_-_Creating_an_Array_Table.mp4
1.00x
03:39/05:49
8:24 PM
4/28/2012

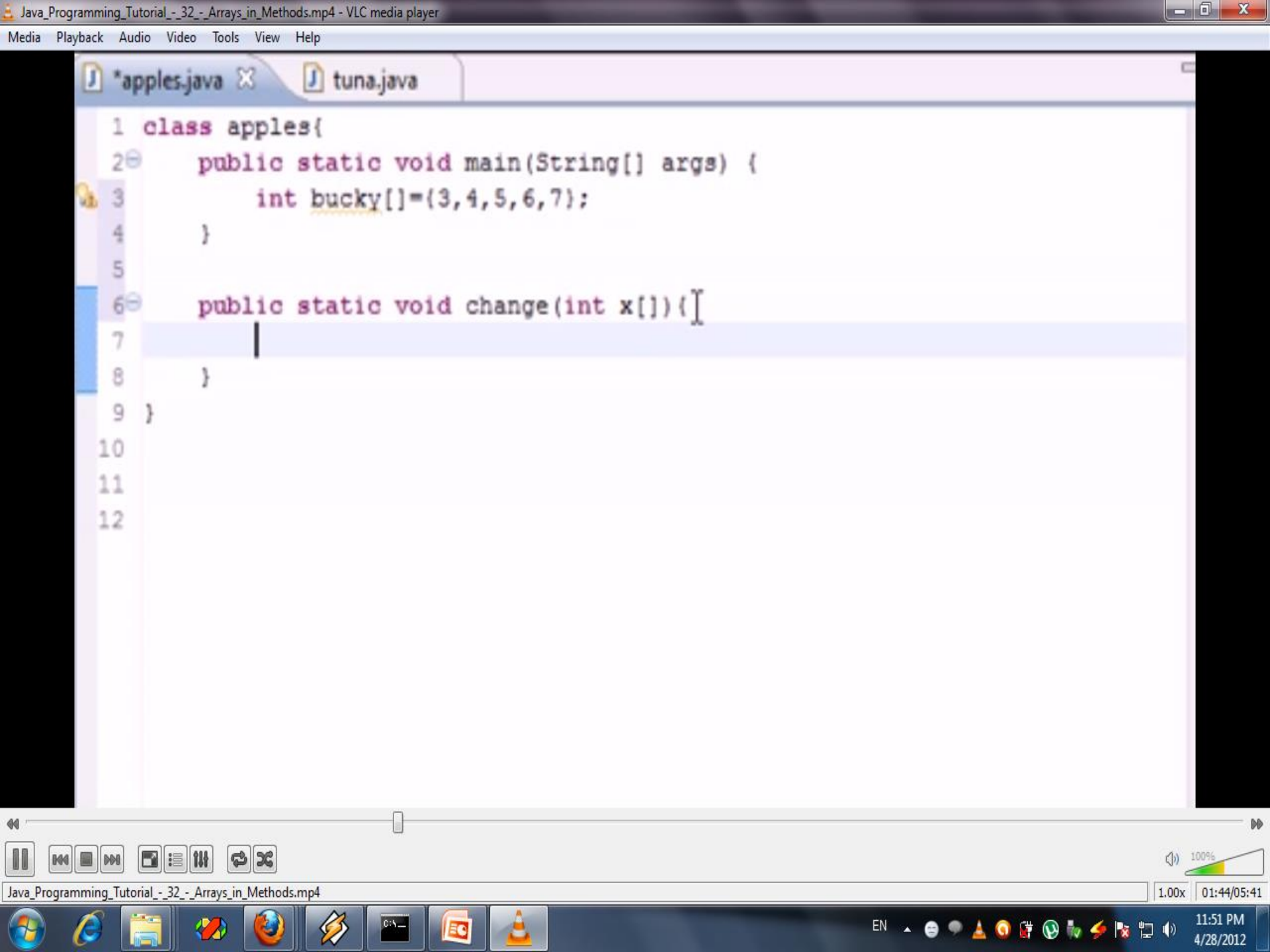

Java_Programming_Tutorial_-_28_-_Creating_an_Array_Table.mp4 - VLC media player
Media Playback Audio Video Tools View Help
Problems Javadoc Declaration Console
<terminated> apples [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 10, 2009 10:41:39 PM)
Index Value
0 32
1 12
2 18
3 54
4 2
Java_Programming_Tutorial_-_28_-_Creating_an_Array_Table.mp4
1.00x
03:45/05:49
8:36 PM
4/28/2012

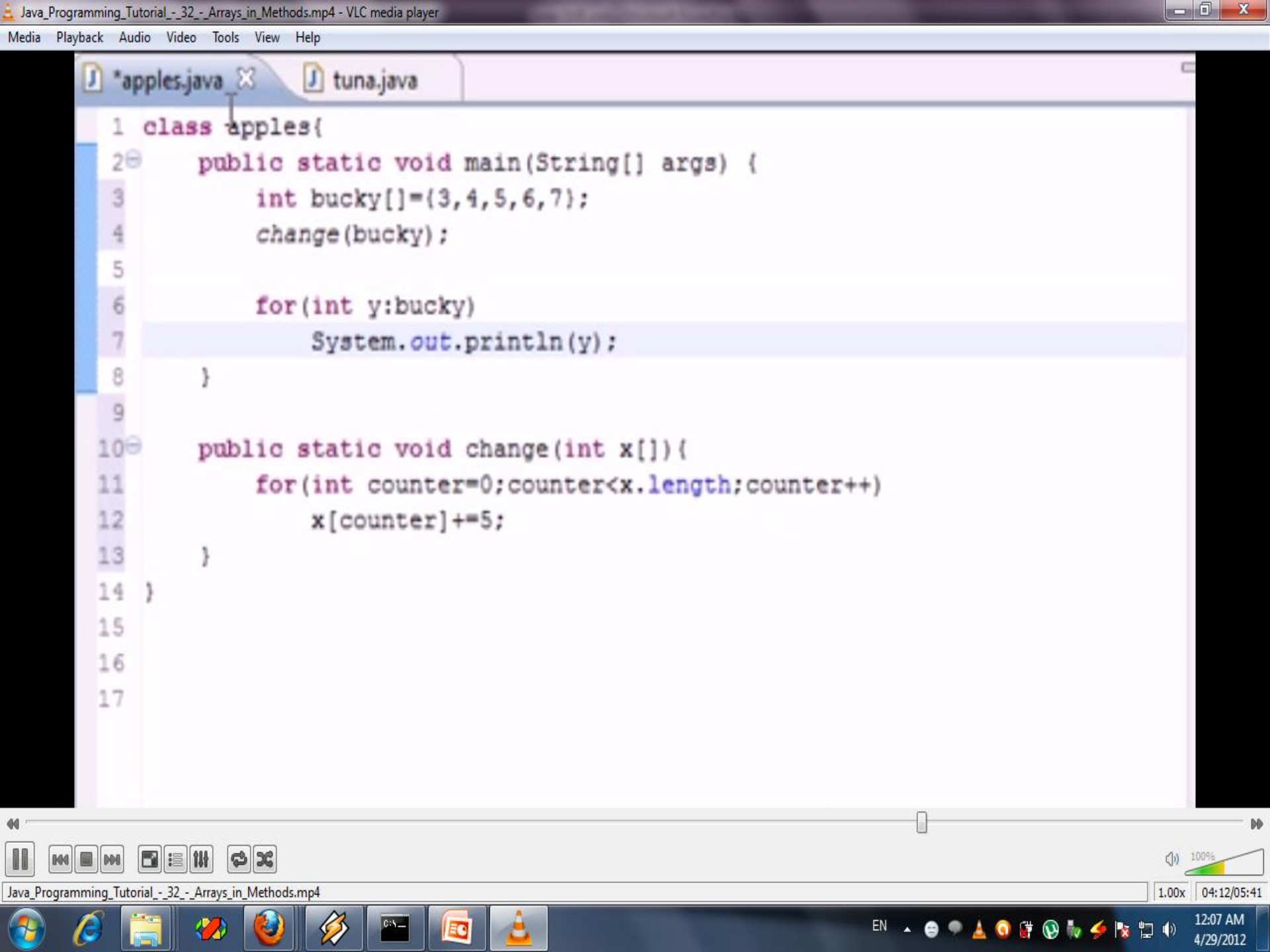
Sum up the elements of an array

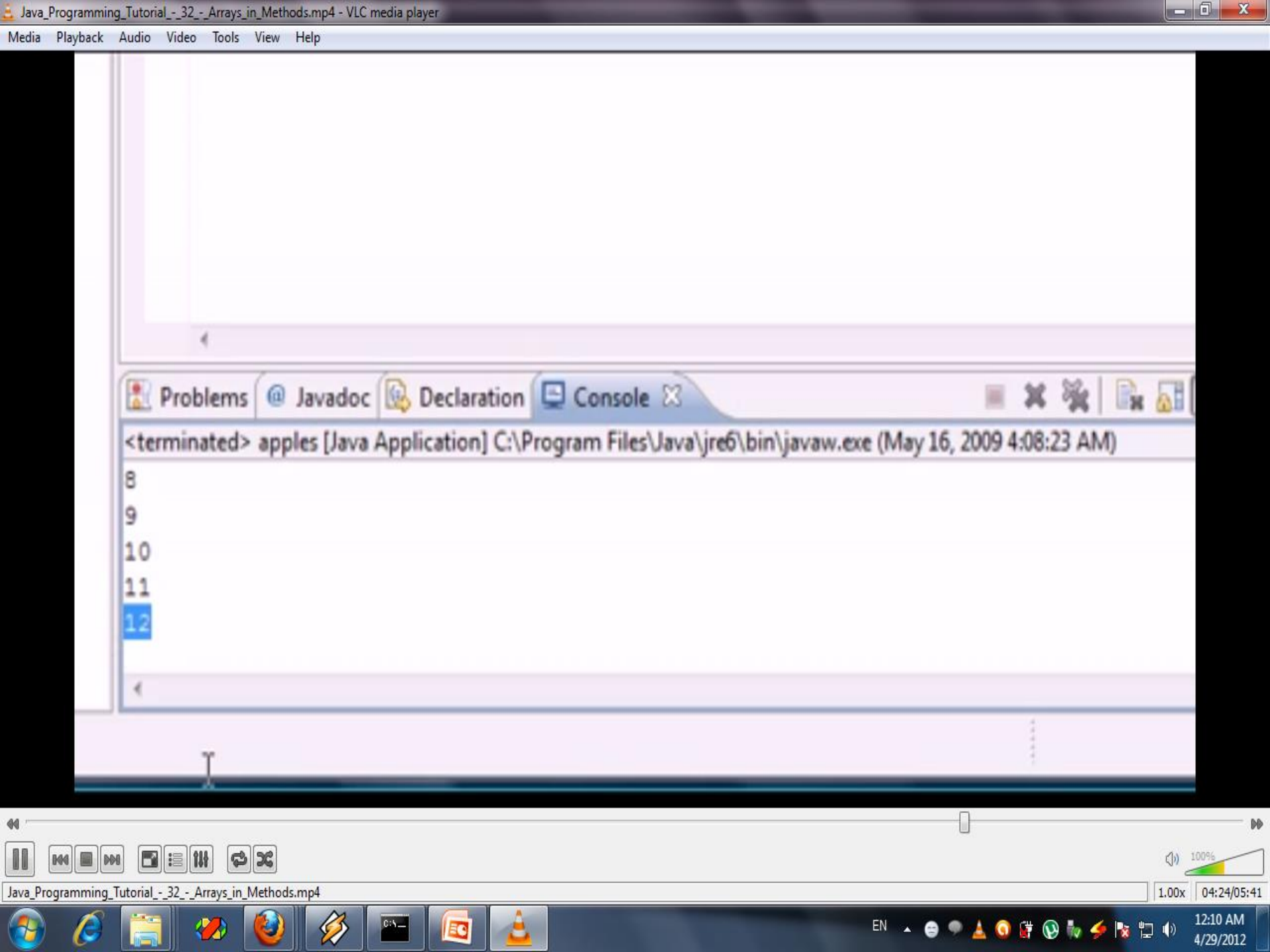

Java_Programming_Tutorial_-_29_-_Summing_Elements_of_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
class apples{
public static void main(String[] args) {
int bucky[]={21,16,86,21,3};
int sum=0;
for(int counter=0;counter<bucky.length;counter++){
sum+=bucky[counter];
}
System.out.println("The sum of theze numbers is " +sum);
}
}
Java_Programming_Tutorial_-_29_-_Summing_Elements_of_Arrays.mp4
1.00x
02:59/04:00
8:49 PM
4/28/2012


Java_Programming_Tutorial_-_29_-_Summing_Elements_of_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
Problems Javadoc Declaration Console
<terminated> apples [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 15, 2009 11:00:51 PM)
The sum of theze numbers is 147
Java_Programming_Tutorial_-_29_-_Summing_Elements_of_Arrays.mp4
1.00x
03:02/04:00
8:50 PM
4/28/2012

How to use arrays with methods(how to pass an array to a method, and catch up that array in the method)


Java_Programming_Tutorial_-_32_-_Arrays_in_Methods.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3 int bucky[]={3,4,5,6,7};
4 }
5
6 public static void change(int x[]){
7
8 }
9 }
10
11
12
Java_Programming_Tutorial_-_32_-_Arrays_in_Methods.mp4
1.00x
01:44/05:41
EN
11:51 PM
4/28/2012


Java_Programming_Tutorial_-_32_-_Arrays_in_Methods.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3 int bucky[]={3,4,5,6,7};
4 change(bucky);
5
6 for(int y:bucky)
7 System.out.println(y);
8 }
9
10 public static void change(int x[]){
11 for(int counter=0;counter<x.length;counter++)
12 x[counter]+=5;
13 }
14 }
15
16
17
Java_Programming_Tutorial_-_32_-_Arrays_in_Methods.mp4
1.00x
04:12/05:41
EN
12:07 AM
4/29/2012

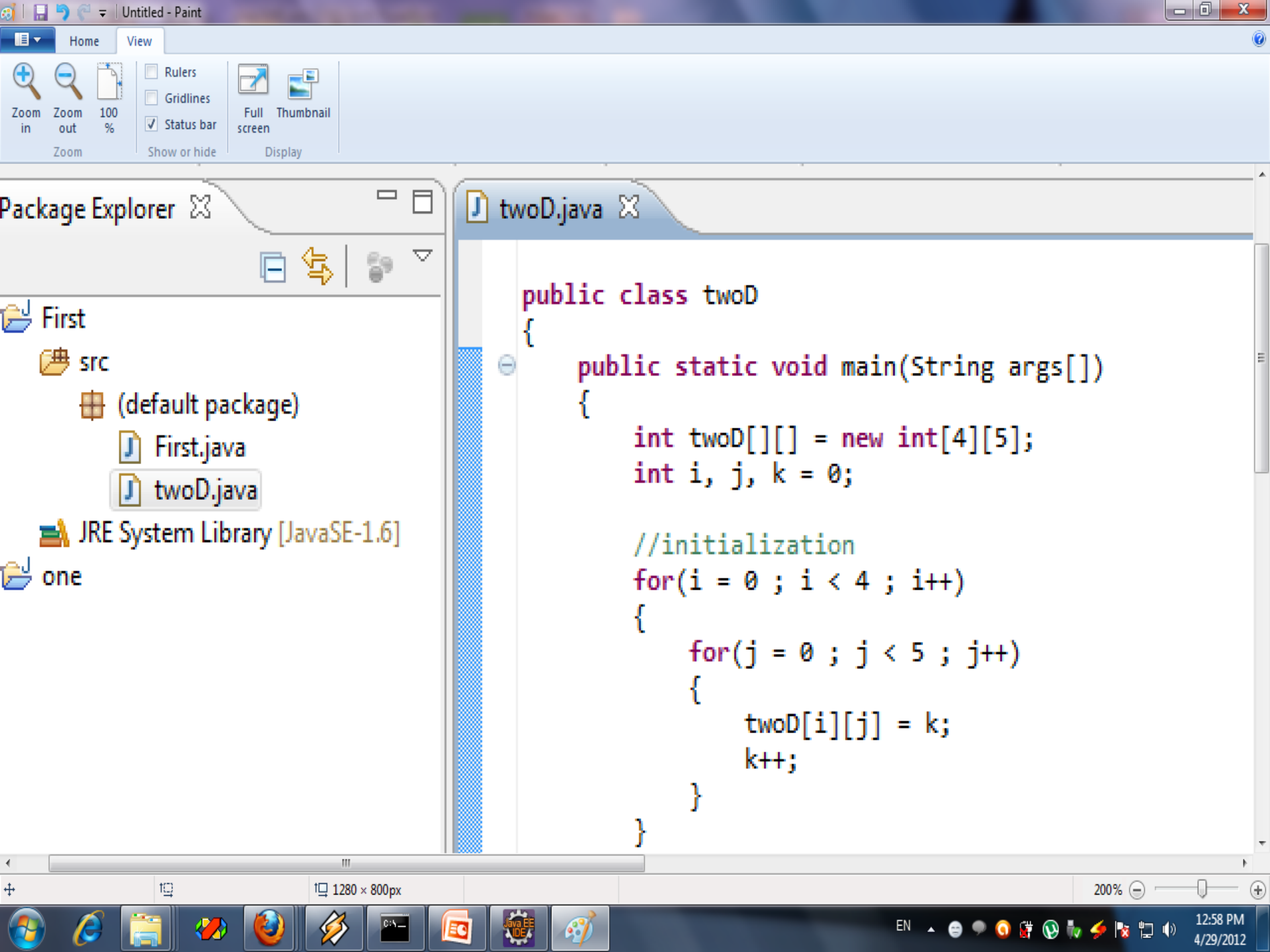

Java_Programming_Tutorial_-_32_-_Arrays_in_Methods.mp4 - VLC media player
Media Playback Audio Video Tools View Help
Problems Javadoc Declaration Console
<terminated> apples [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 16, 2009 4:08:23 AM)
8
9
10
11
12
Java_Programming_Tutorial_-_32_-_Arrays_in_Methods.mp4
1.00x
04:24/05:41
EN
12:10 AM
4/29/2012

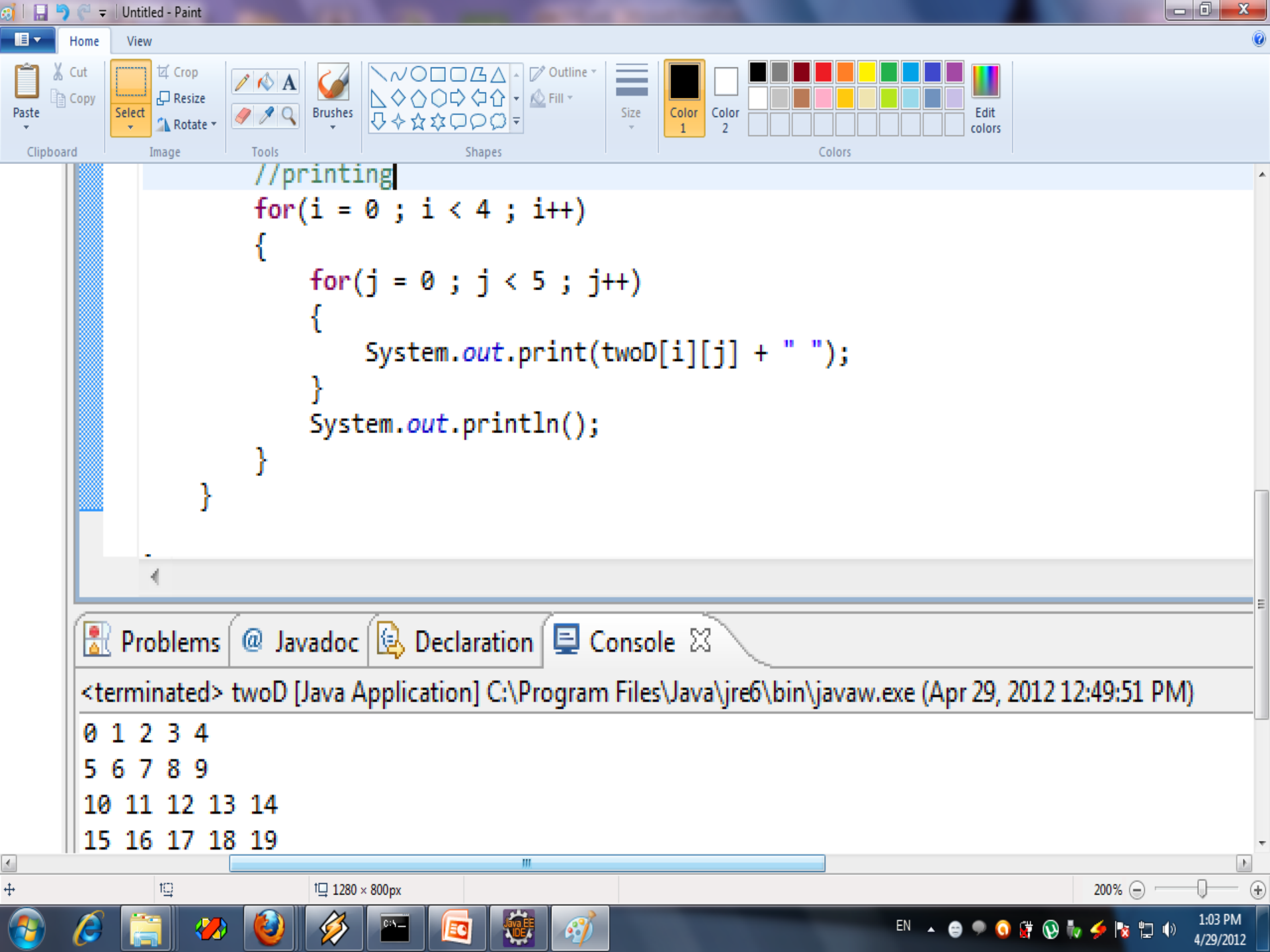
Multidimensional Arrays

- Multidimensional arrays are '*array of arrays*'.
- Declaration, creation and initialization of multidimensional arrays are almost same like one-dimensional arrays with some small differences.
- The implementation of multi dimensional array has two pair square brackets after the array name, if it is a two dimensional array.

Multidimensional Arrays

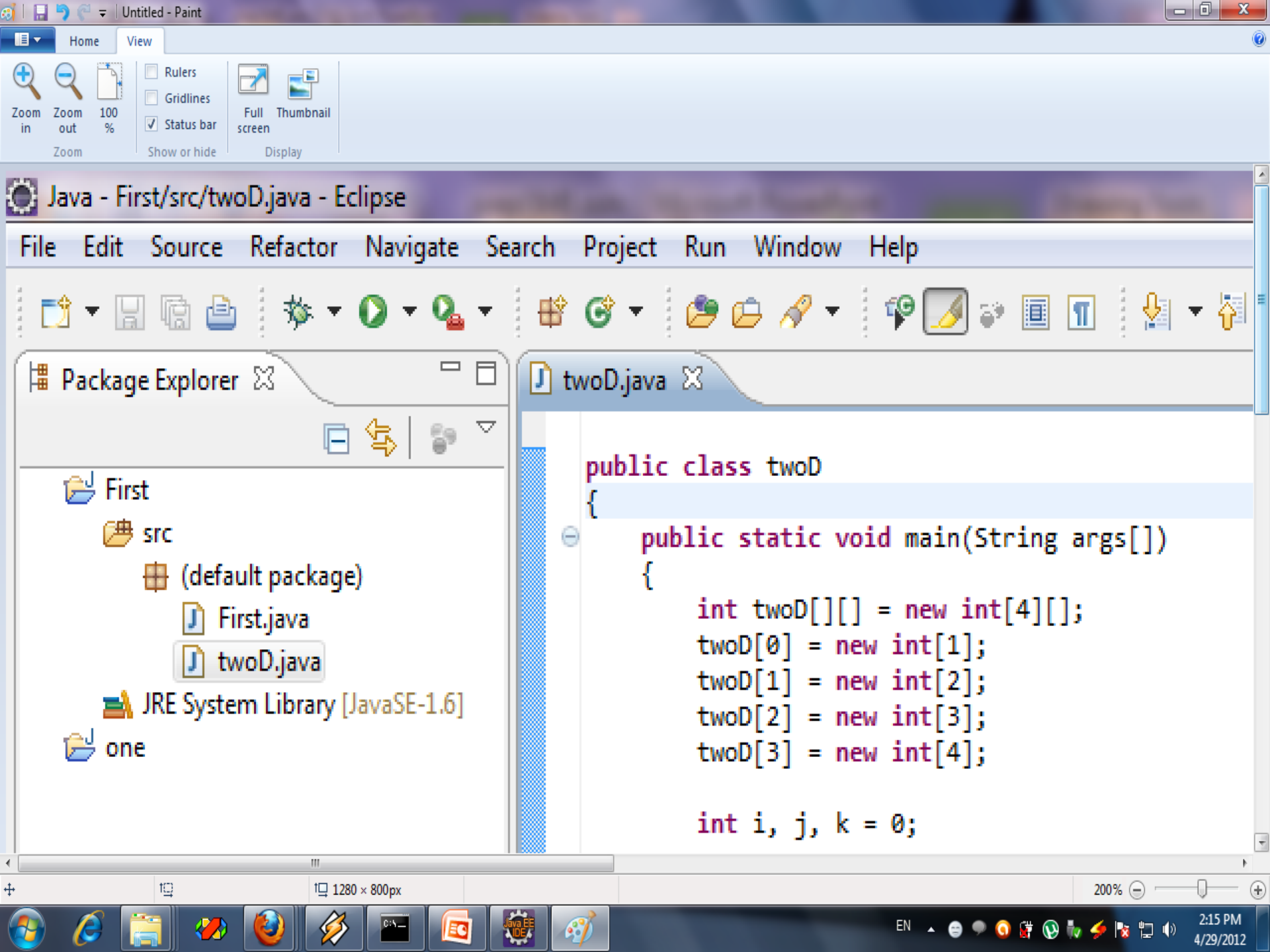
- Following is a declaration of two-dimensional array :
  - int twoDarray[ ][ ] = new int[4][5];
  - Similarly, int[ ][ ] twoDarray = new int[4][5];
  - A two-dimensional array is created here, where, there are 4 rows and 5 columns.
  - Here, an array of arrays is created if int datatype.
  - Following is an example of initialization and then displaying the values of a two-dimensional array:

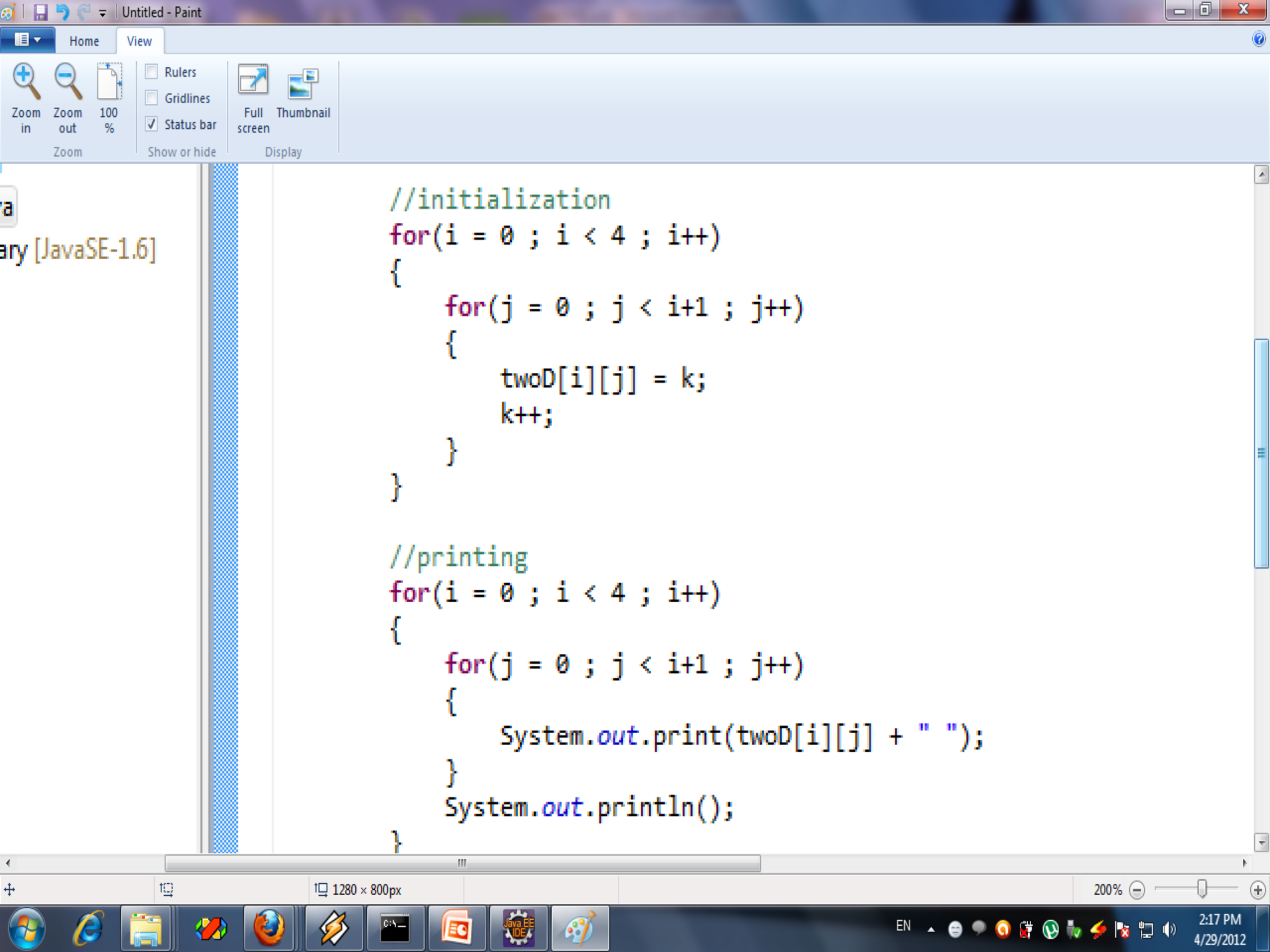

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Package Explorer
First
src
(default package)
First.java
twoD.java
JRE System Library [JavaSE-1.6]
one
twoD.java
public class twoD
{
public static void main(String args[])
{
int twoD[][] = new int[4][5];
int i, j, k = 0;
//initialization
for(i = 0 ; i < 4 ; i++)
{
for(j = 0 ; j < 5 ; j++)
{
twoD[i][j] = k;
k++;
}
}
1280 × 800px
200%
EN
12:58 PM
4/29/2012

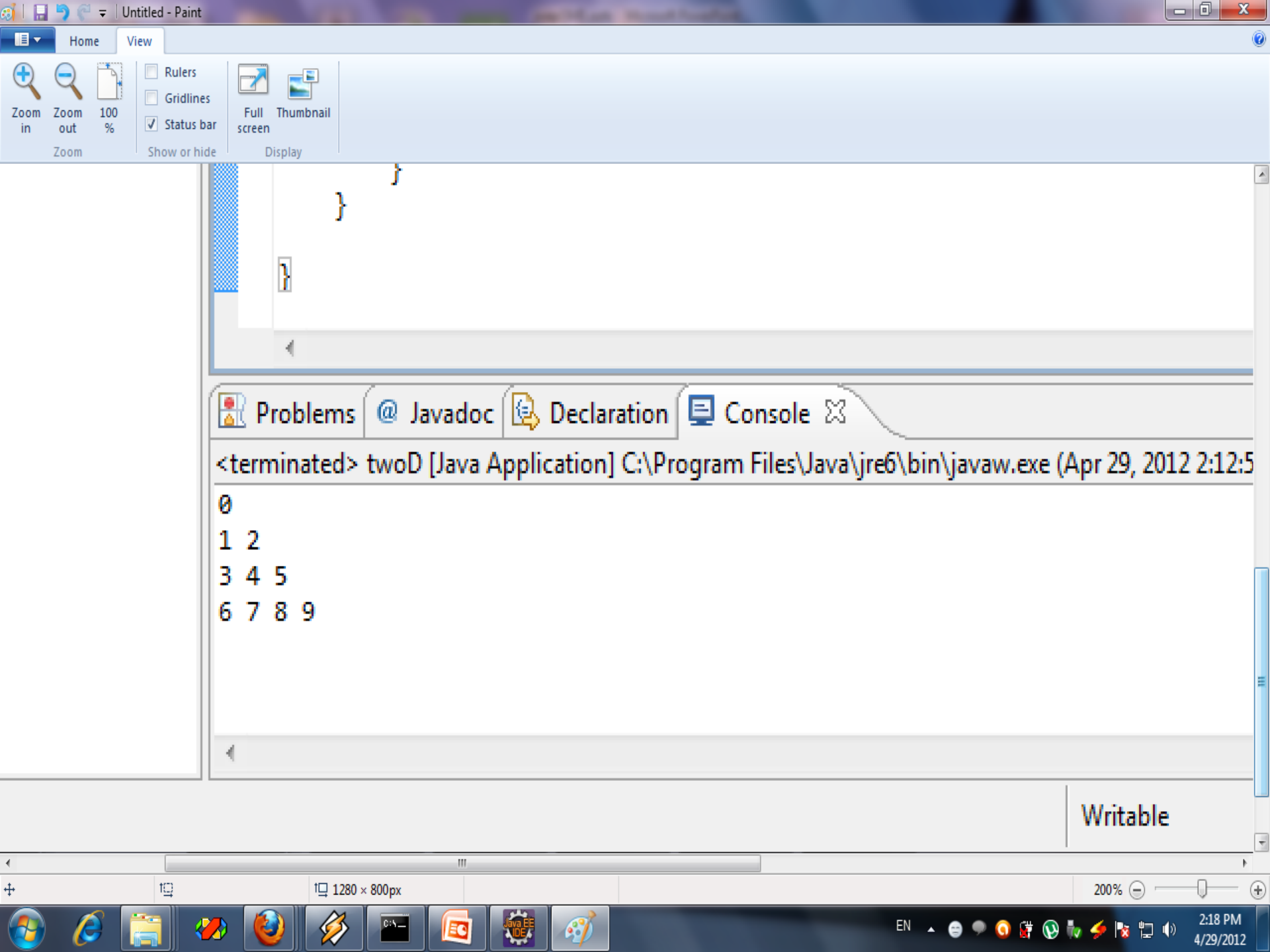

Untitled - Paint
Home
View
Cut
Copy
Paste
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Clipboard
Image
Tools
Shapes
Colors
//printing
for(i = 0 ; i < 4 ; i++)
{
for(j = 0 ; j < 5 ; j++)
{
System.out.print(twoD[i][j] + " ");
}
System.out.println();
}
}
Problems
Javadoc
Declaration
Console
<terminated> twoD [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (Apr 29, 2012 12:49:51 PM)
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
15 16 17 18 19
1280 × 800px
200%
EN
1:03 PM
4/29/2012

Multidimensional arrays

- Another method of declaring two-dimensional array:
  - We can take the size of arrays under the upper level array in our control.
  - int twoD[ ][ ] = new int[4][ ];
  - A two-dimensional array is created here, where, there are 4 rows, but the number of columns in each row is variable.

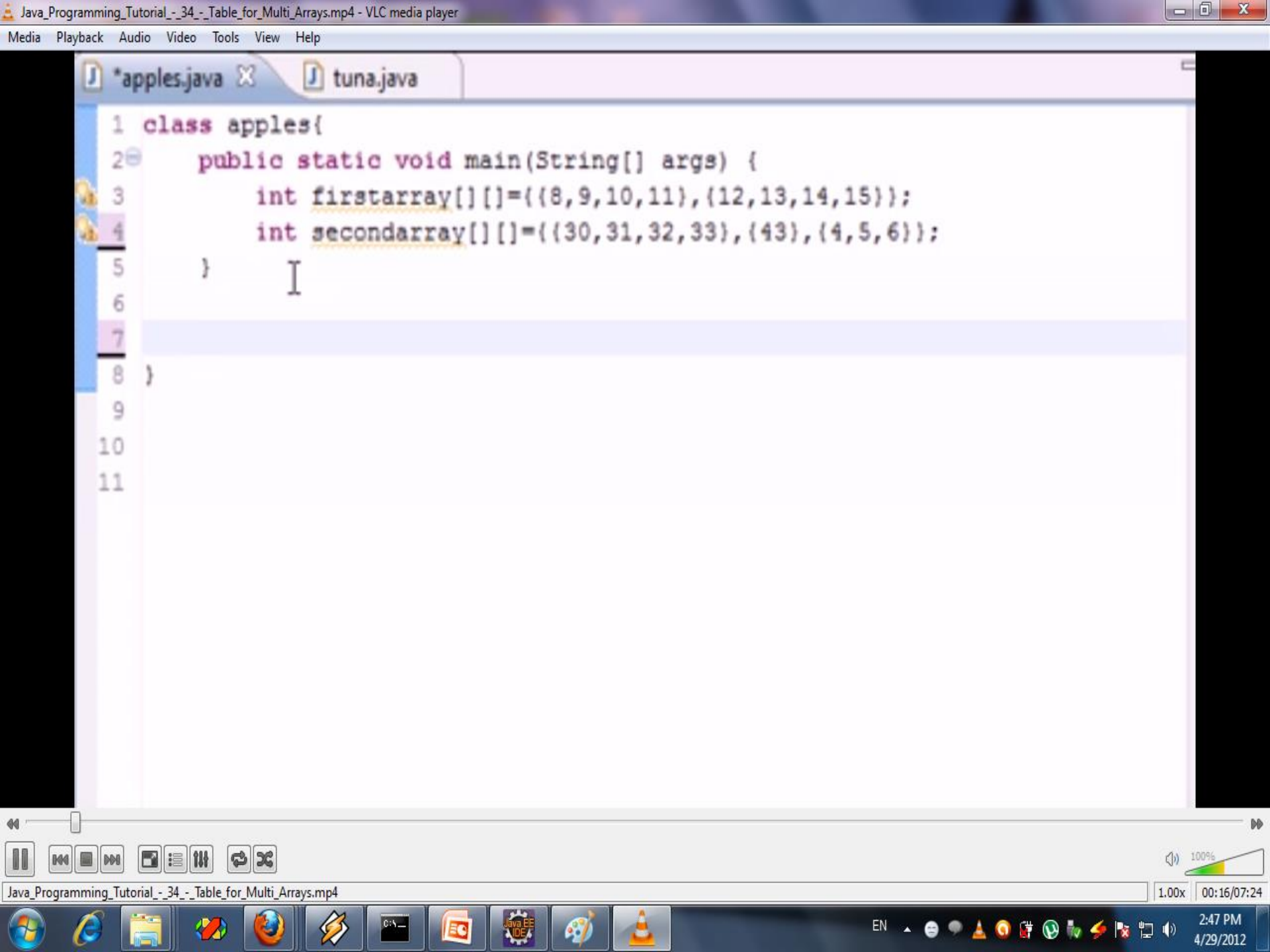

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Java - First/src/twoD.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
First
src
(default package)
First.java
twoD.java
JRE System Library [JavaSE-1.6]
one
twoD.java
public class twoD
{
public static void main(String args[])
{
int twoD[][] = new int[4][];
twoD[0] = new int[1];
twoD[1] = new int[2];
twoD[2] = new int[3];
twoD[3] = new int[4];
int i, j, k = 0;
1280 × 800px
200%
EN
2:15 PM
4/29/2012

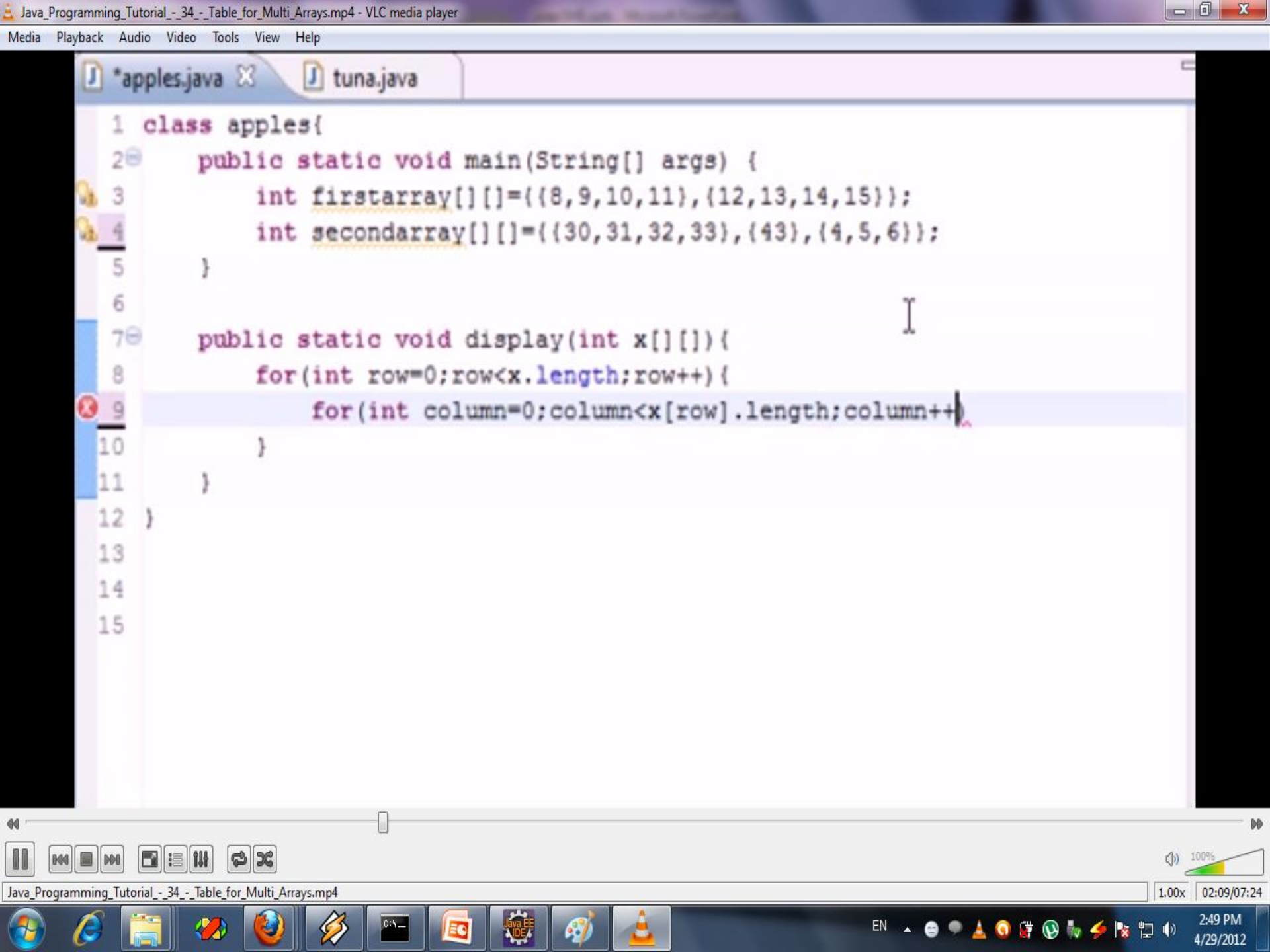

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
ary [JavaSE-1.6]
//initialization
for(i = 0 ; i < 4 ; i++)
{
for(j = 0 ; j < i+1 ; j++)
{
twoD[i][j] = k;
k++;
}
}
//printing
for(i = 0 ; i < 4 ; i++)
{
for(j = 0 ; j < i+1 ; j++)
{
System.out.print(twoD[i][j] + " ");
}
System.out.println();
}
1280 × 800px
200%
EN
2:17 PM
4/29/2012

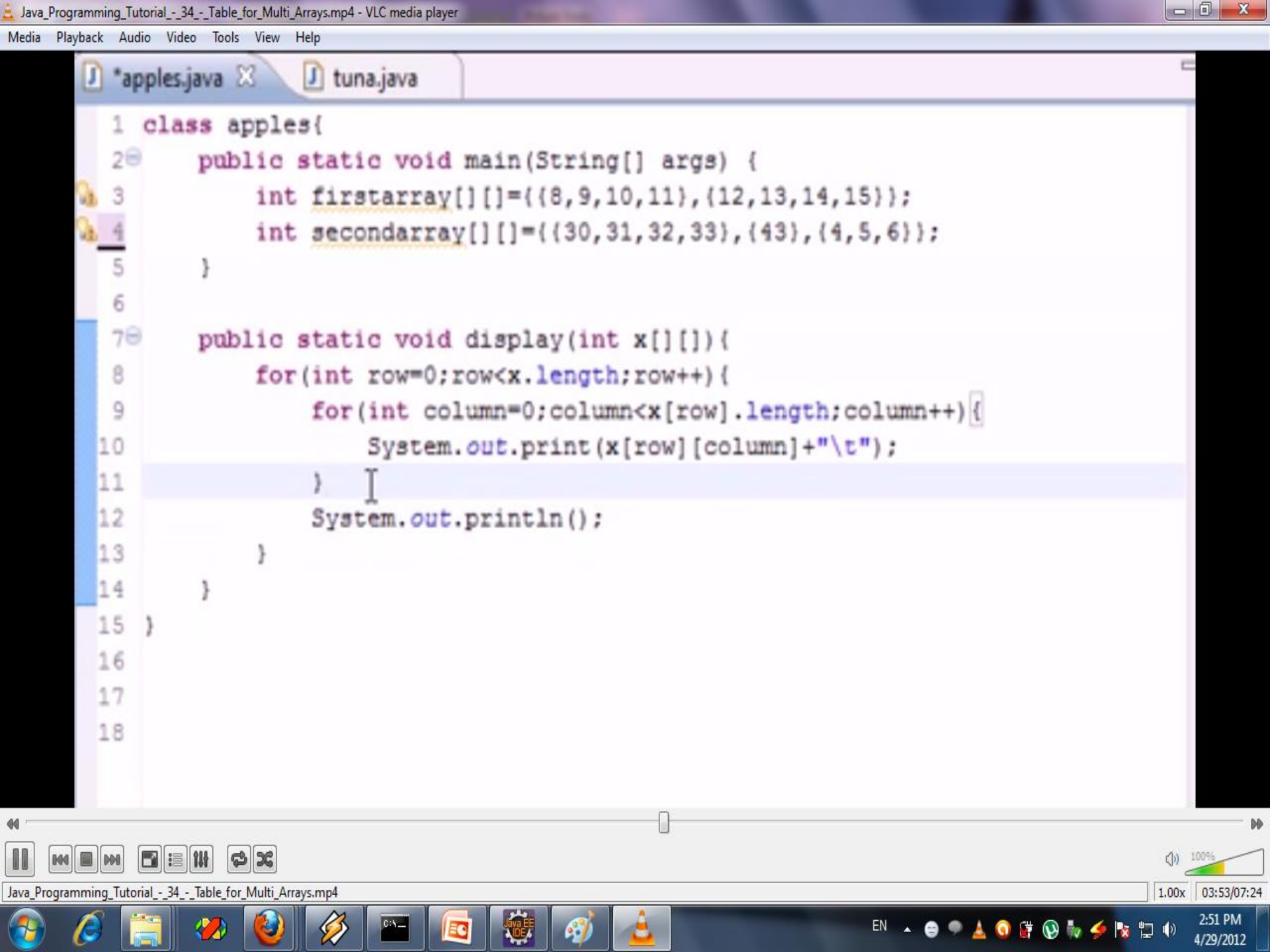

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
}
}
}
Problems
Javadoc
Declaration
Console
<terminated> twoD [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (Apr 29, 2012 2:12:5
0
1 2
3 4 5
6 7 8 9
Writable
1280 × 800px
200%
EN
2:18 PM
4/29/2012

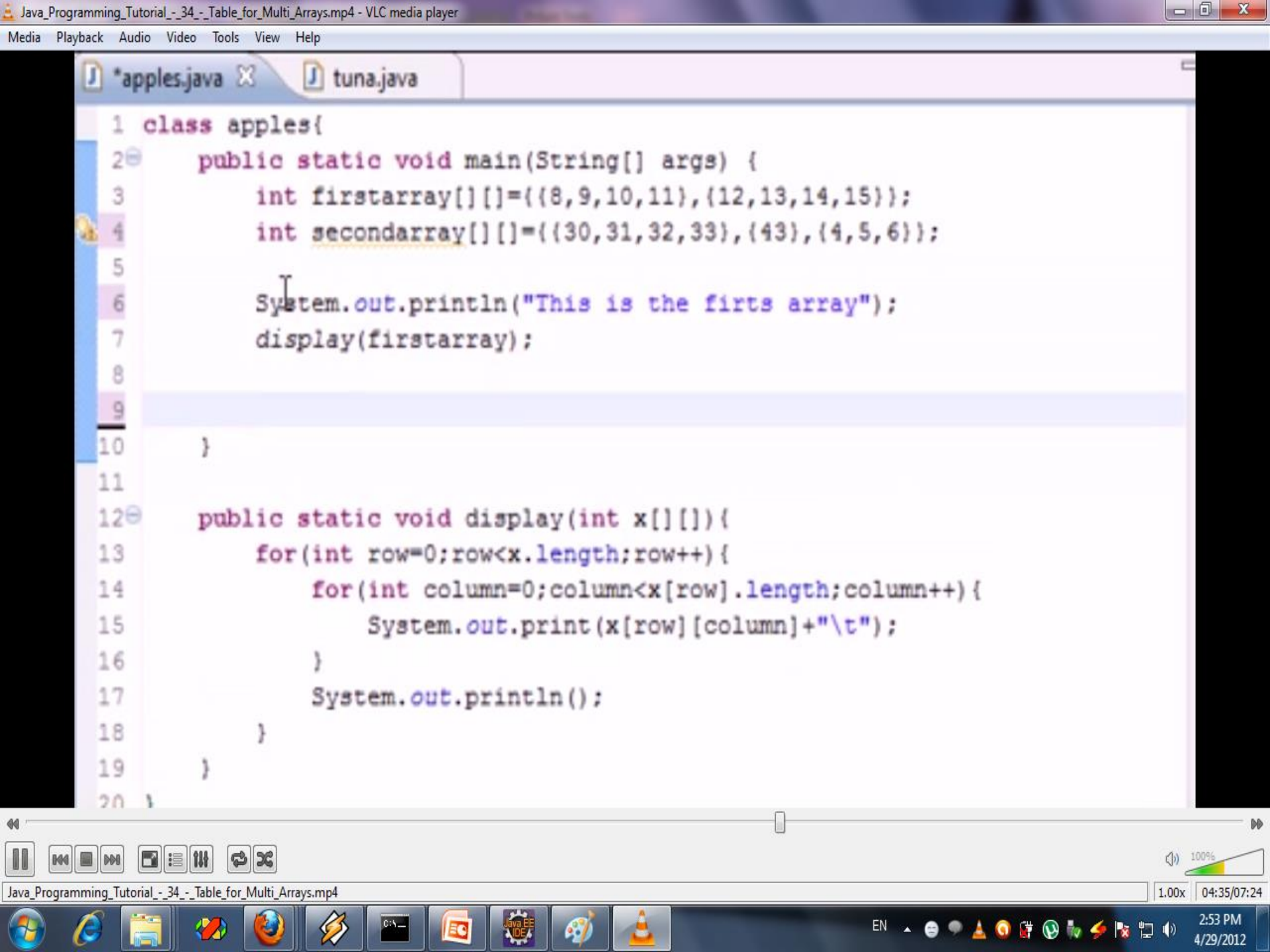
Two-dimensional arrays in method

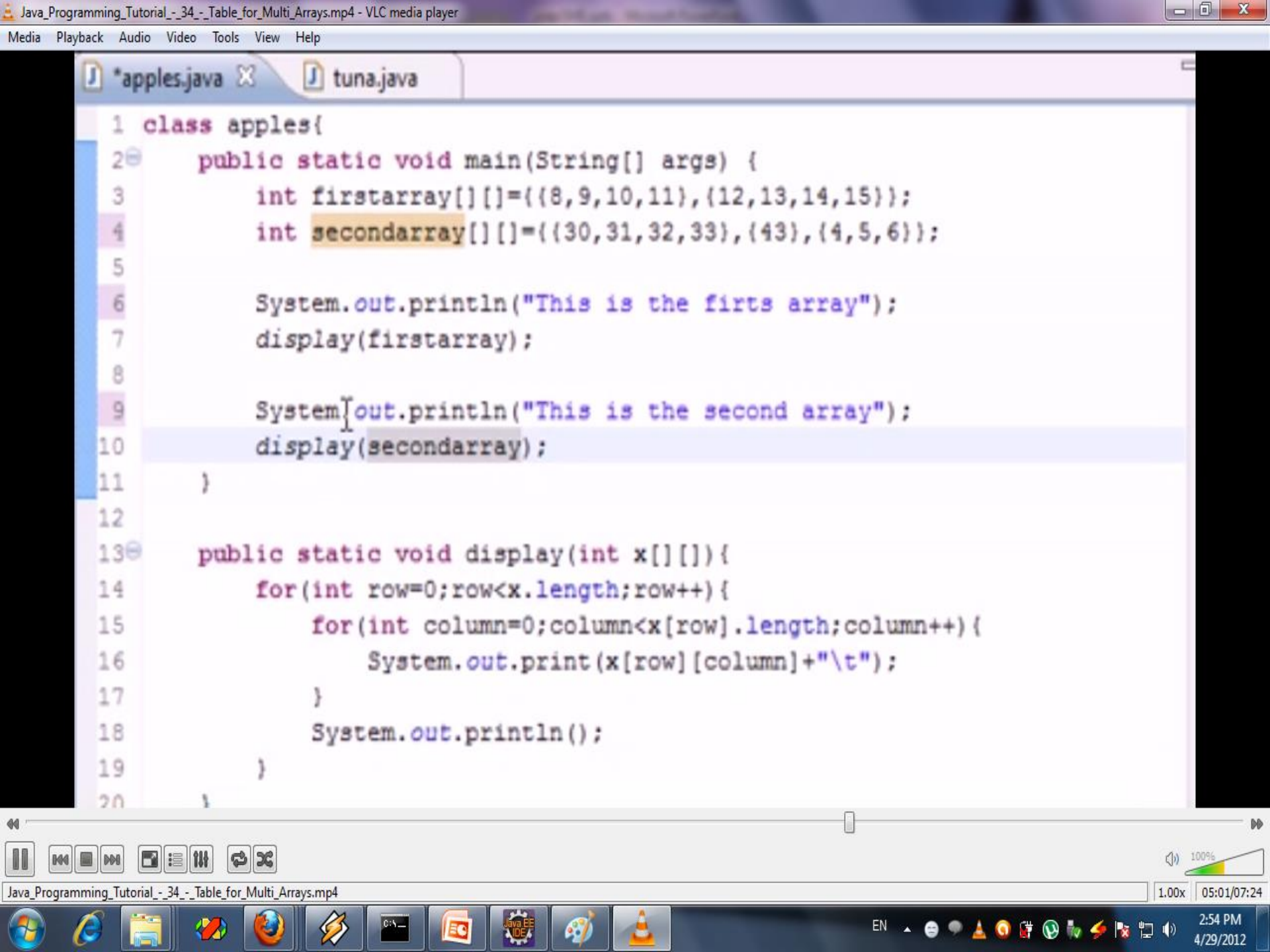
- This is the last example that we will study about arrays.
- Here, we will see couple of new things:
  - Another way of initialization of two-dimensional array
  - Passing two dimensional array as argument while calling another method.


Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3 int firstarray[][]={{8,9,10,11},{12,13,14,15}};
4 int secondarray[][]={{30,31,32,33},{43},{4,5,6}};
5 }
6
7
8 }
9
10
11
Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4
1.00x
00:16/07:24
EN
2:47 PM
4/29/2012


Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3 int firstarray[][]={{8,9,10,11},{12,13,14,15}};
4 int secondarray[][]={{30,31,32,33},{43},{4,5,6}};
5 }
6
7 public static void display(int x[][]){
8 for(int row=0;row<x.length;row++){
9 for(int column=0;column<x[row].length;column++)
10 }
11 }
12 }
13
14
15
Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4
1.00x
02:09/07:24
EN
2:49 PM
4/29/2012


Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3 int firstarray[][]={{8,9,10,11},{12,13,14,15}};
4 int secondarray[][]={{30,31,32,33},{43},{4,5,6}};
5 }
6
7 public static void display(int x[][]){
8 for(int row=0;row<x.length;row++){
9 for(int column=0;column<x[row].length;column++){
10 System.out.print(x[row][column]+"\t");
11 }
12 System.out.println();
13 }
14 }
15 }
16
17
18
Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4
1.00x
03:53/07:24
EN
2:51 PM
4/29/2012


Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
class apples{
public static void main(String[] args) {
int firstarray[][]={{8,9,10,11},{12,13,14,15}};
int secondarray[][]={{30,31,32,33},{43},{4,5,6}};
System.out.println("This is the firts array");
display(firstarray);
}
public static void display(int x[][]){
for(int row=0;row<x.length;row++){
for(int column=0;column<x[row].length;column++){
System.out.print(x[row][column]+"\t");
}
System.out.println();
}
}
}
Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4
1.00x
04:35/07:24
EN
2:53 PM
4/29/2012


Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4 - VLC media player
Media Playback Audio Video Tools View Help
*apples.java
tuna.java
1 class apples{
2 public static void main(String[] args) {
3 int firstarray[][]={{8,9,10,11},{12,13,14,15}};
4 int secondarray[][]={{30,31,32,33},{43},{4,5,6}};
5
6 System.out.println("This is the firts array");
7 display(firstarray);
8
9 System.out.println("This is the second array");
10 display(secondarray);
11 }
12
13 public static void display(int x[][]){
14 for(int row=0;row<x.length;row++){
15 for(int column=0;column<x[row].length;column++){
16 System.out.print(x[row][column]+"\t");
17 }
18 System.out.println();
19 }
20 }
Java_Programming_Tutorial_-_34_-_Table_for_Multi_Arrays.mp4
1.00x
05:01/07:24
EN
2:54 PM
4/29/2012

```
20      }
21  }
22
23
24
```

Problems | Javadoc | Declaration | Console

<terminated> apples [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 16, 2009 11:12:46 AM)

```
This is the fifts array
8       9       10      11
12      13      14      15
This is the second array
30      31      32      33
43
```

Inheritance

Inheritance

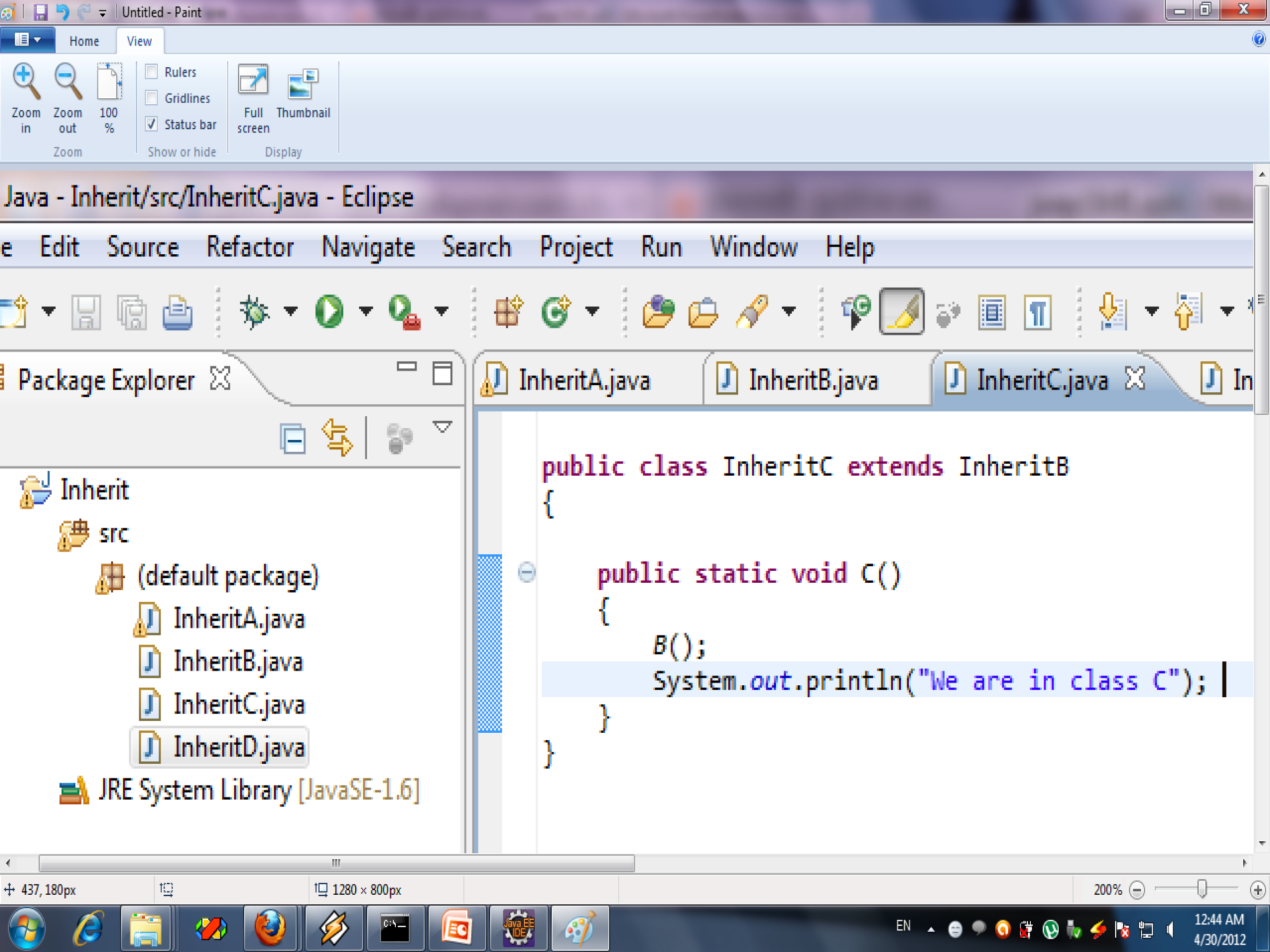
- Inheritance can be defined as the process where one object acquires the properties of another.
- With the use of inheritance, the information is made manageable in a hierarchical order.
- A class that is inherited is called a ***superclass***.
- The class that does the inheriting is called a ***subclass***.

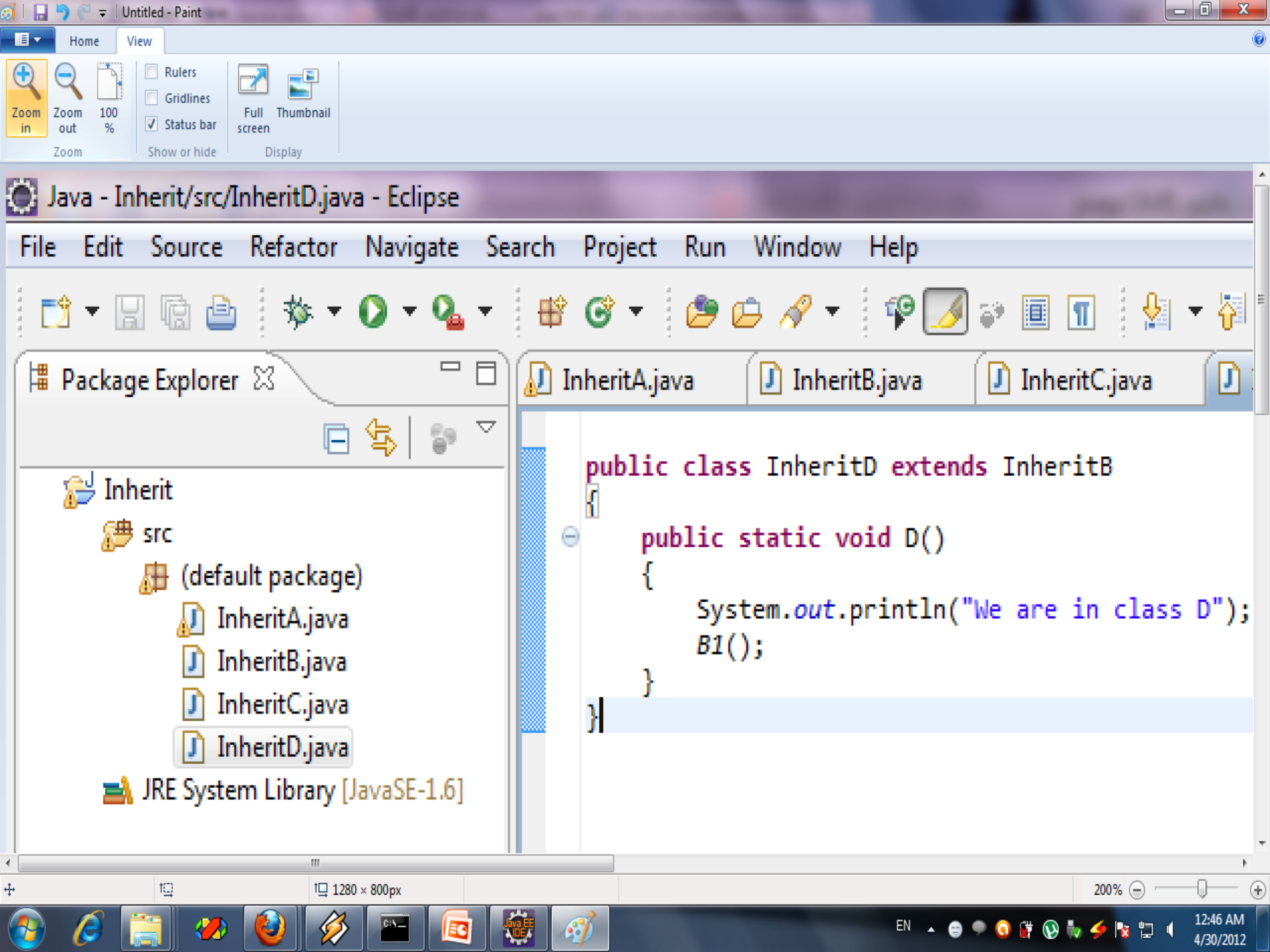
Inheritance

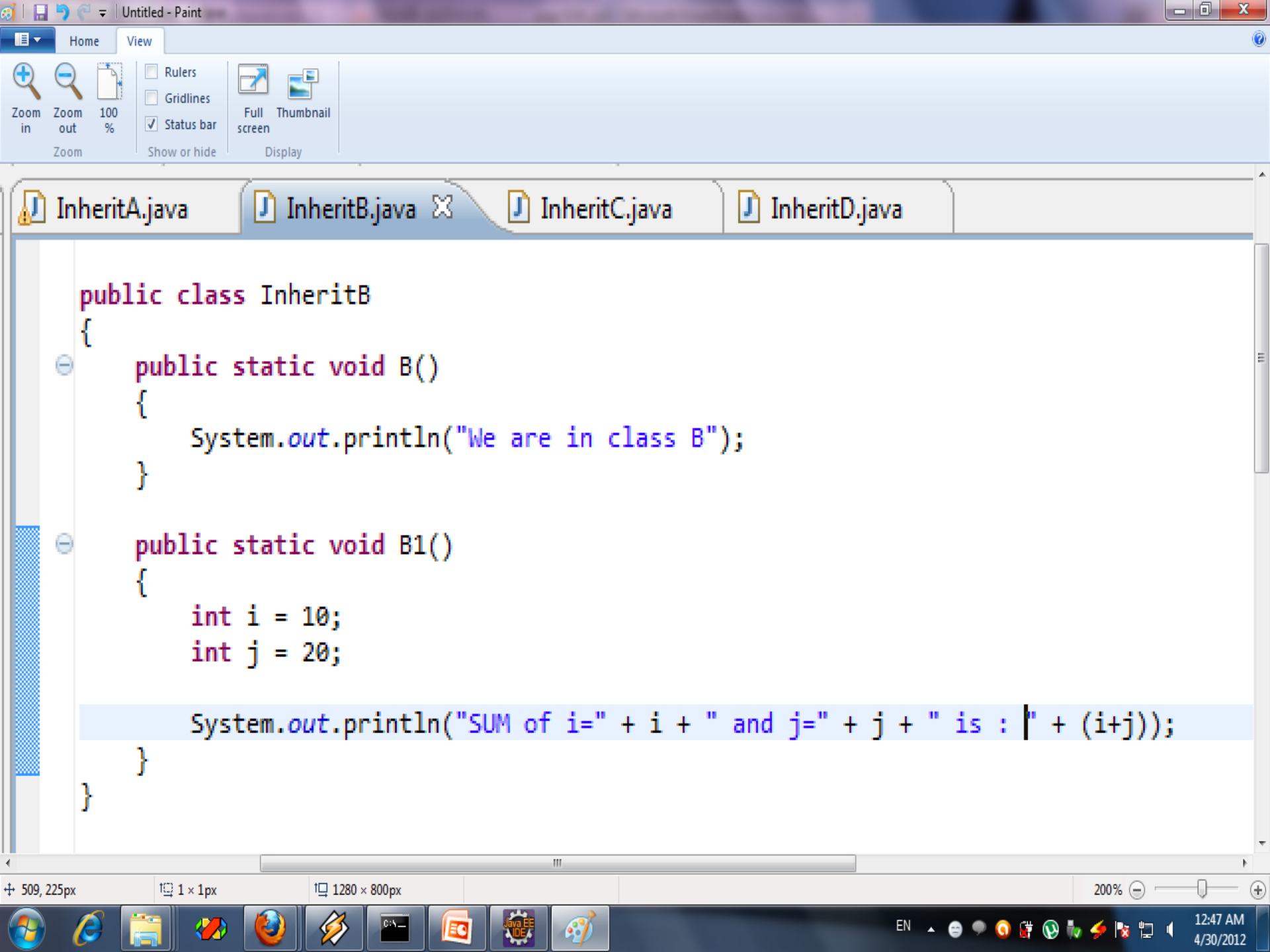
- A subclass is a specialized version of a superclass.
- The subclass inherits all of the instance variables and methods defined by the superclass and add with its own.
- To inherit a class, simply incorporate the definition of one class into another by using the **extends** keyword.

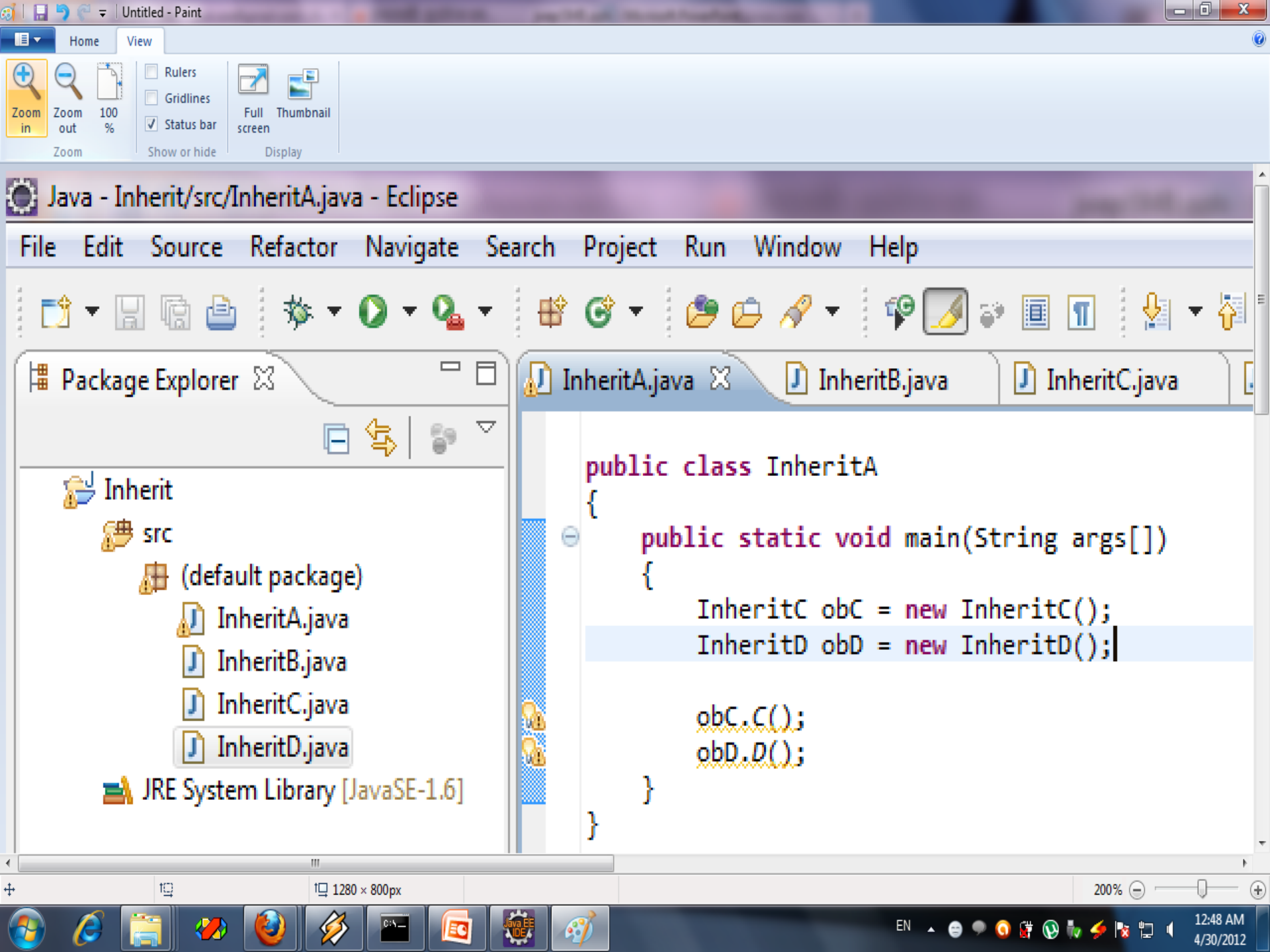
Inheritance

- The first and simple example of inheritance:
  - There four classes(A,B,C,D) in the example.
  - Class A contains the main function
  - Class C and D both inherits Class B
  - So, here the “superclass B” has two “subclasses C and D”.


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Java - Inherit/src/InheritC.java - Eclipse
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
Package Explorer
Inherit
src
(default package)
InheritA.java
InheritB.java
InheritC.java
InheritD.java
JRE System Library [JavaSE-1.6]
InheritA.java
InheritB.java
InheritC.java
public class InheritC extends InheritB
{
public static void C()
{
B();
System.out.println("We are in class C");
}
}
437, 180px
1280 × 800px
200%
EN
12:44 AM
4/30/2012

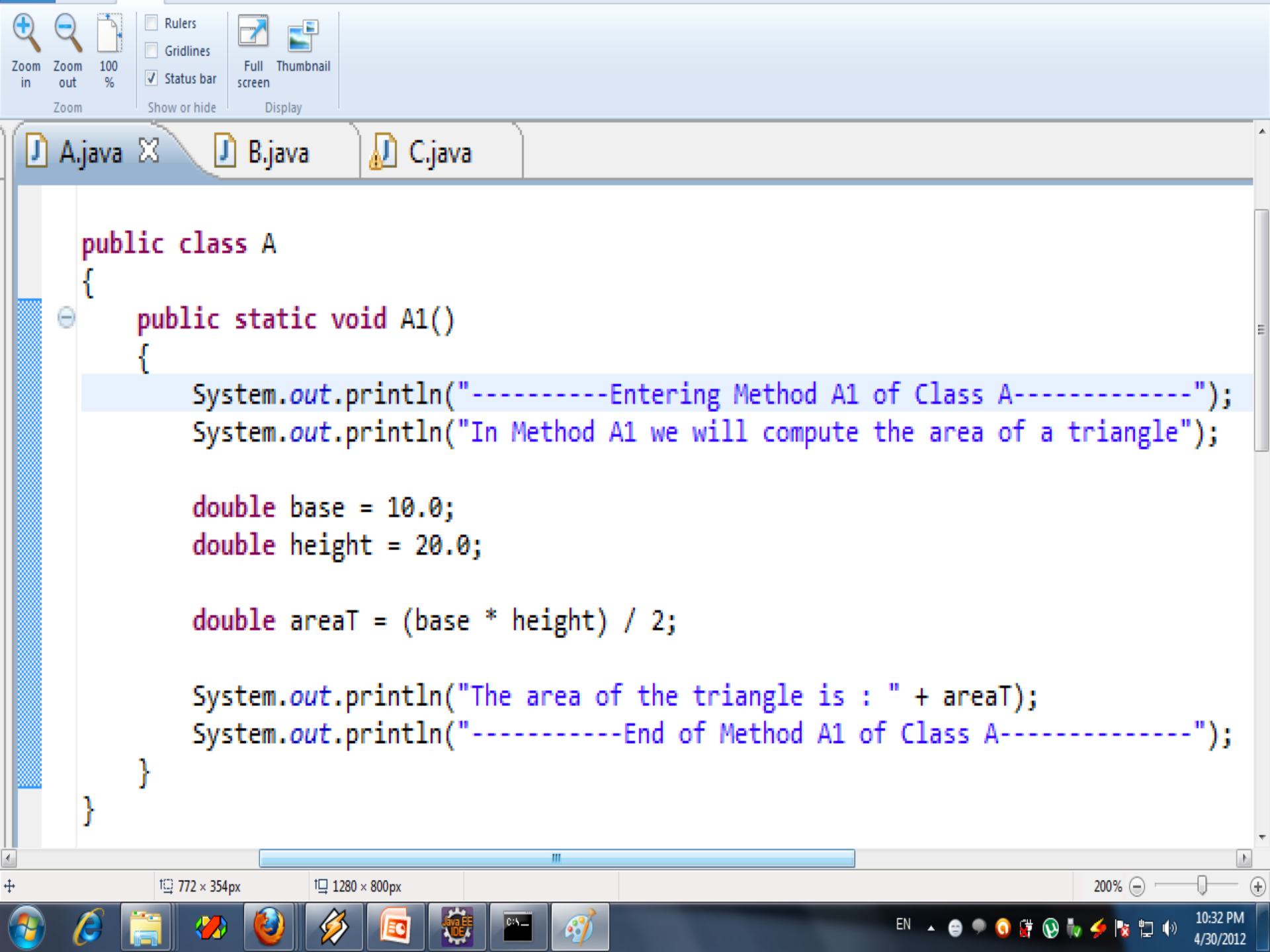

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Java - Inherit/src/InheritD.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Inherit
src
(default package)
InheritA.java
InheritB.java
InheritC.java
InheritD.java
JRE System Library [JavaSE-1.6]
InheritA.java
InheritB.java
InheritC.java
public class InheritD extends InheritB
{
public static void D()
{
System.out.println("We are in class D");
B1();
}
}
1280 × 800px
200%
EN
12:46 AM
4/30/2012

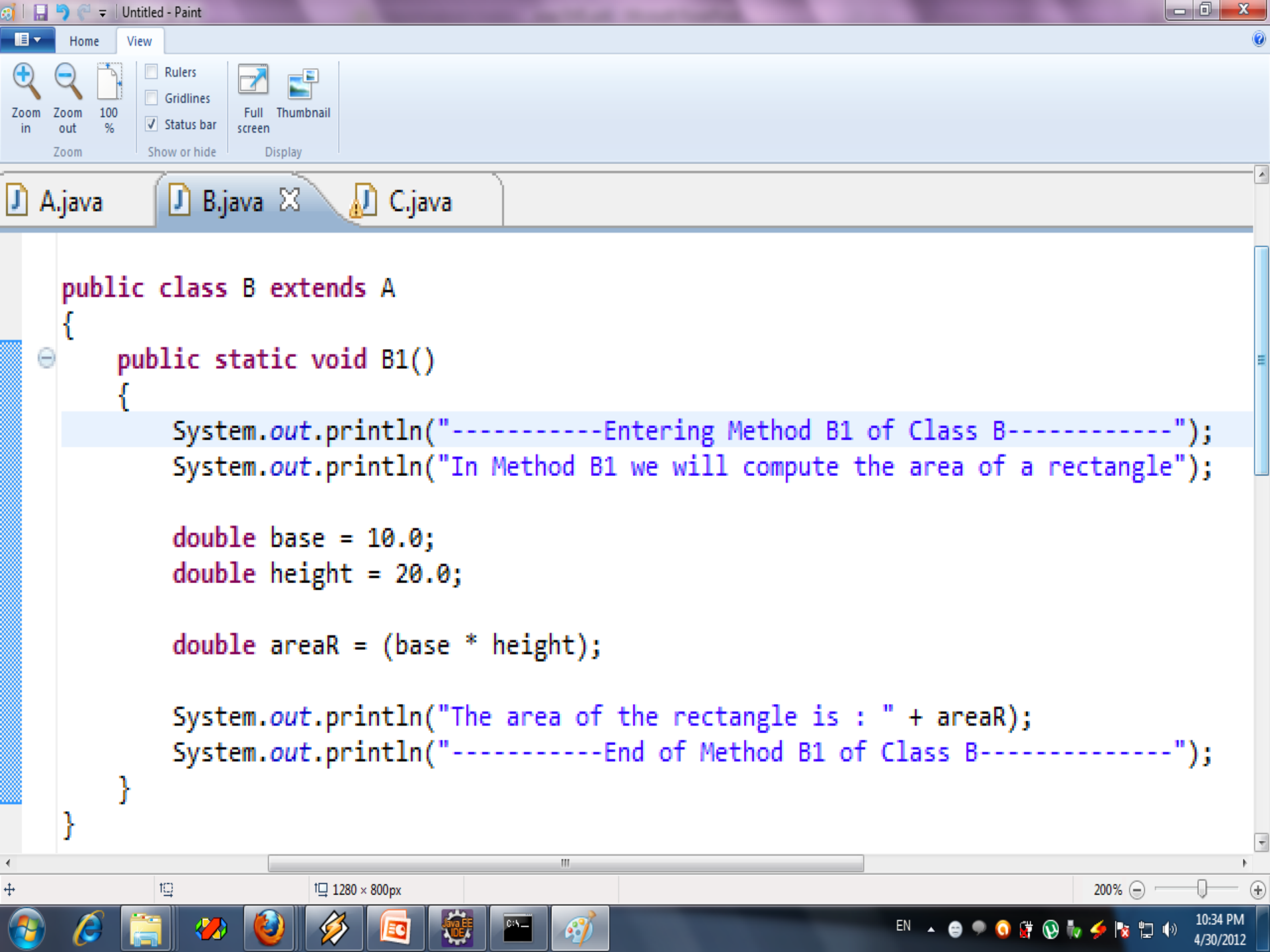

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
InheritA.java
InheritB.java
InheritC.java
InheritD.java
public class InheritB
{
public static void B()
{
System.out.println("We are in class B");
}
public static void B1()
{
int i = 10;
int j = 20;
System.out.println("SUM of i=" + i + " and j=" + j + " is : " + (i+j));
}
}
509, 225px
1 × 1px
1280 × 800px
200%
EN
12:47 AM
4/30/2012

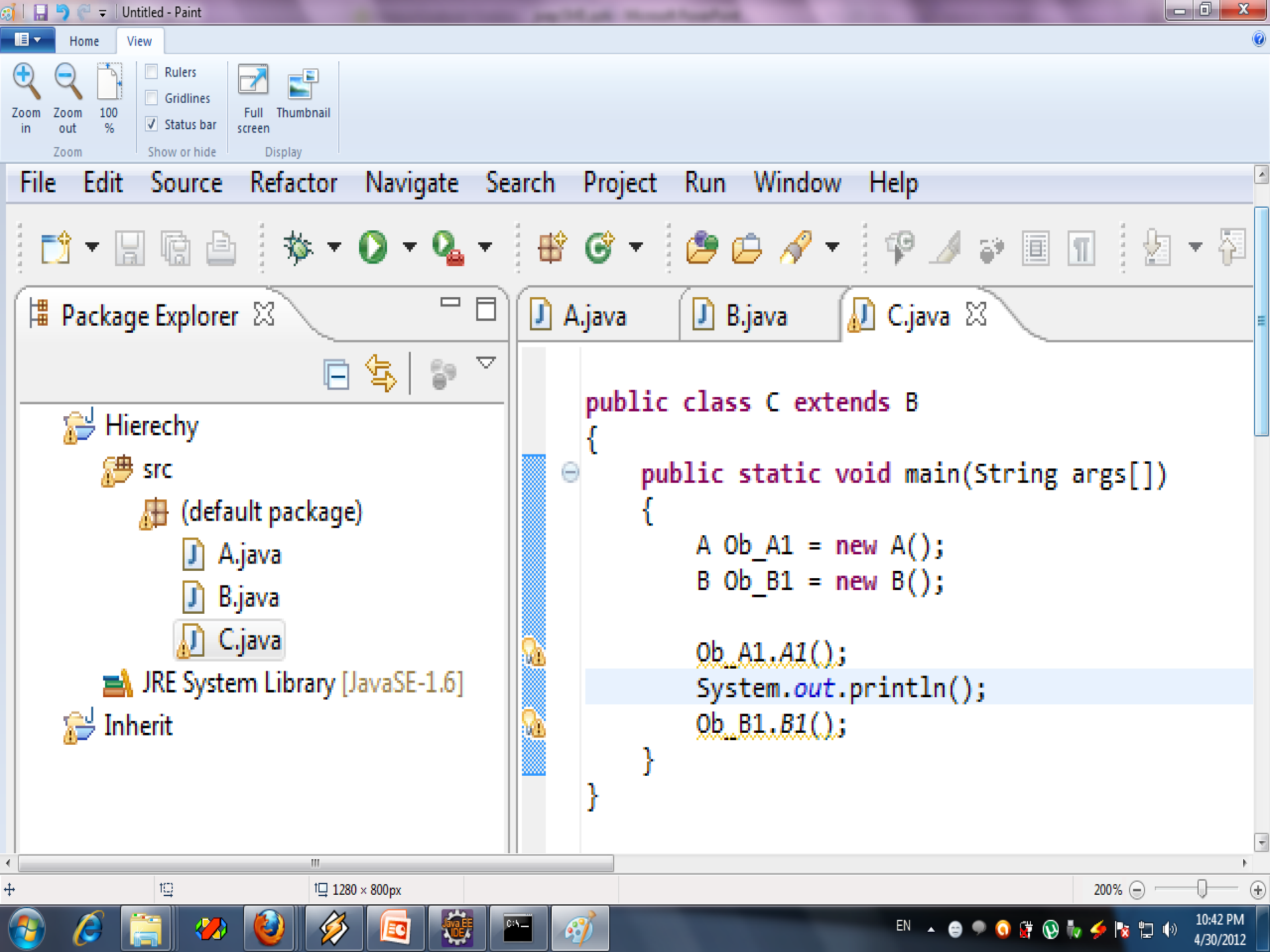

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Java - Inherit/src/InheritA.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Inherit
src
(default package)
InheritA.java
InheritB.java
InheritC.java
InheritD.java
JRE System Library [JavaSE-1.6]
InheritA.java
InheritB.java
InheritC.java
public class InheritA
{
public static void main(String args[])
{
InheritC obC = new InheritC();
InheritD obD = new InheritD();
obC.C();
obD.D();
}
}
1280 × 800px
200%
EN
12:48 AM
4/30/2012

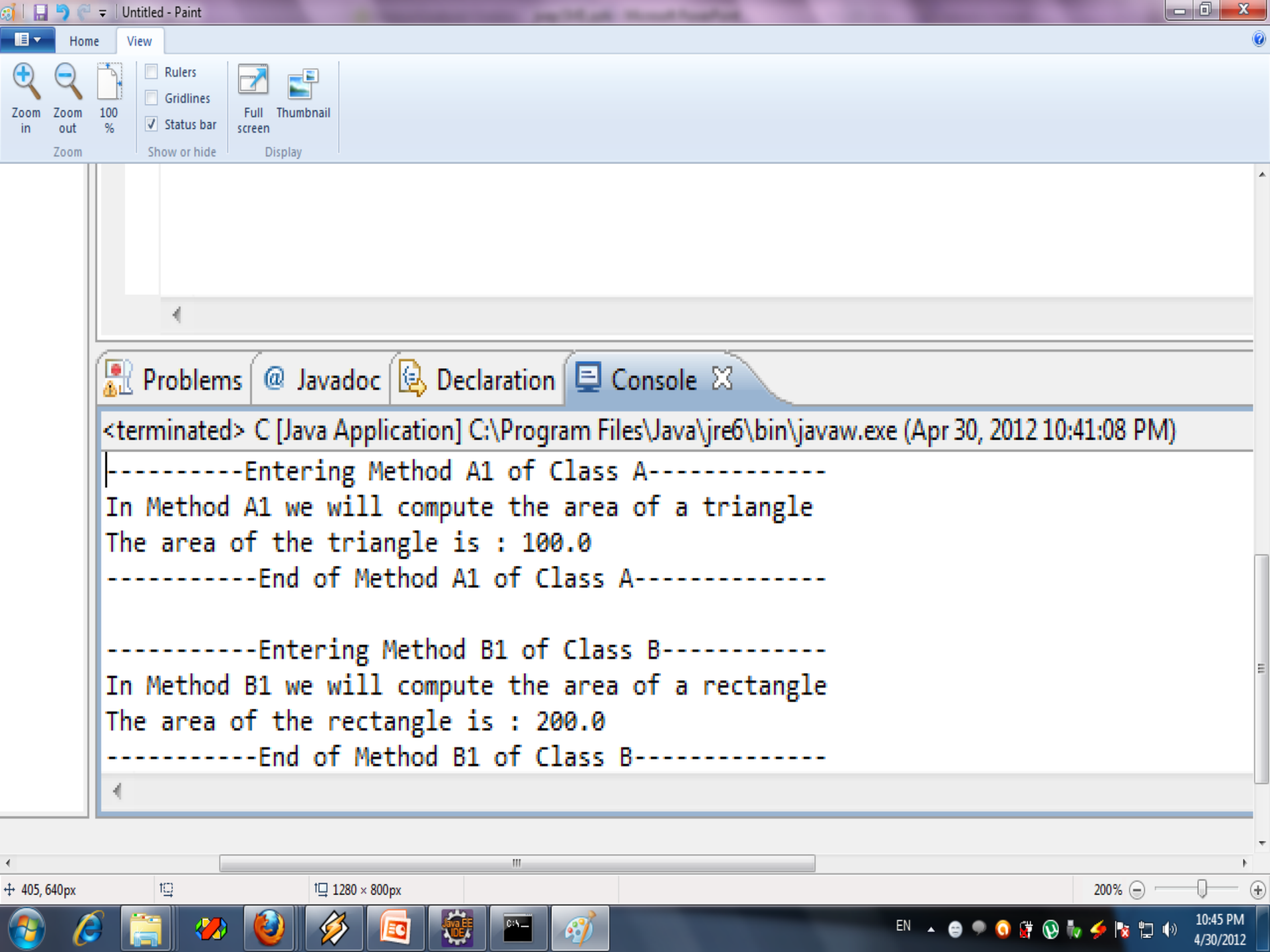
Multilevel Hierarchy of Inheritance

- In the previous example, there was one superclass and multiple(two) subclasses.
- It is also perfectly acceptable to use a subclass as a superclass of another.
- Let, there are 3 classes, A, B and C. B can inherit A and C can inherit B.
- After that, class C inherits both the properties of A and B.


Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
A.java
B.java
C.java
public class A
{
public static void A1()
{
System.out.println("----------Entering Method A1 of Class A-------------");
System.out.println("In Method A1 we will compute the area of a triangle");
double base = 10.0;
double height = 20.0;
double areaT = (base * height) / 2;
System.out.println("The area of the triangle is : " + areaT);
System.out.println("----------End of Method A1 of Class A-------------");
}
}
772 × 354px
1280 × 800px
200%
EN
10:32 PM
4/30/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
A.java
B.java
C.java
public class B extends A
{
public static void B1()
{
System.out.println("----------Entering Method B1 of Class B----------");
System.out.println("In Method B1 we will compute the area of a rectangle");
double base = 10.0;
double height = 20.0;
double areaR = (base * height);
System.out.println("The area of the rectangle is : " + areaR);
System.out.println("----------End of Method B1 of Class B----------");
}
}
1280 × 800px
200%
EN
10:34 PM
4/30/2012

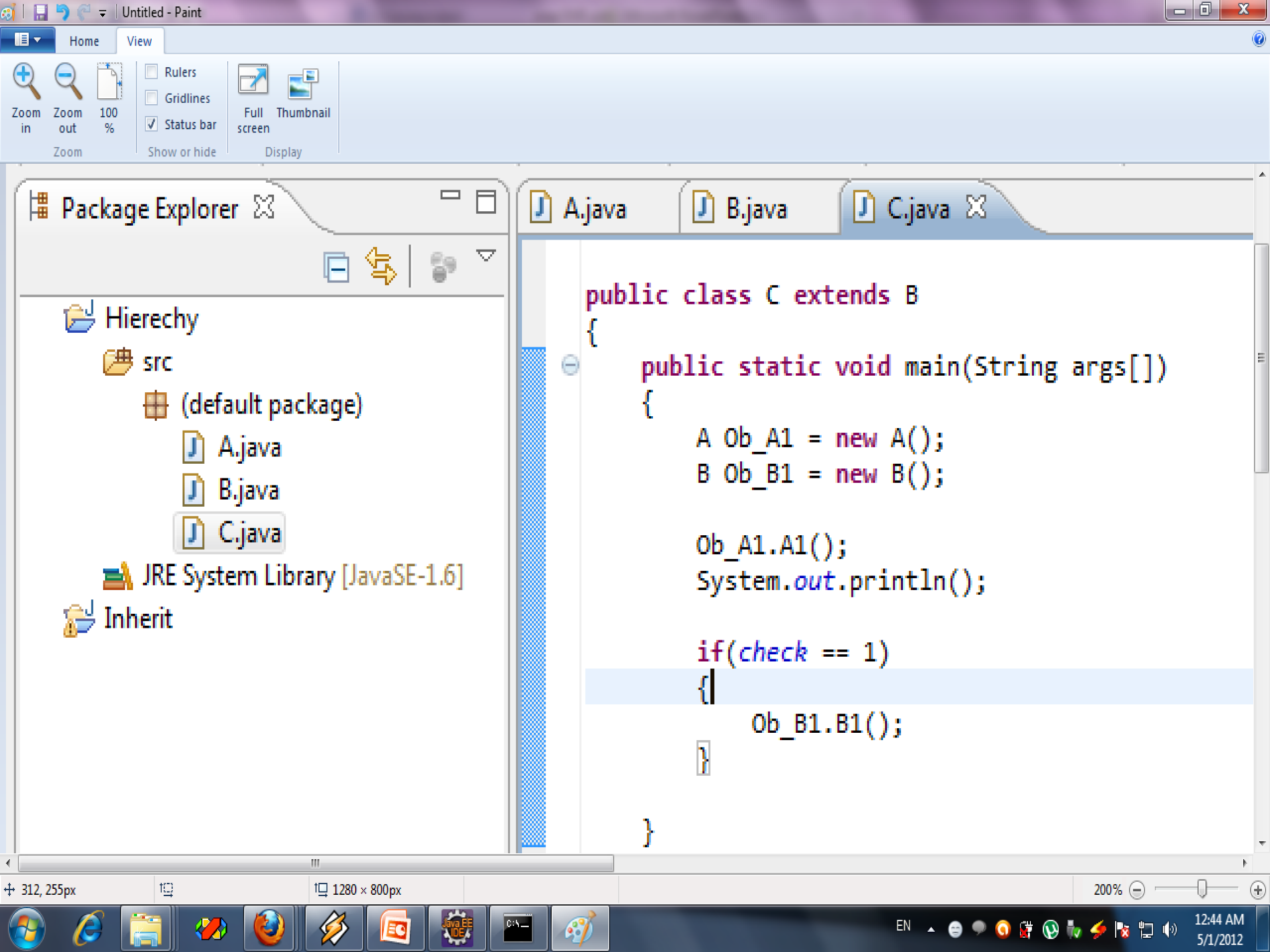

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Hierechy
src
(default package)
A.java
B.java
C.java
JRE System Library [JavaSE-1.6]
Inherit
A.java
B.java
C.java
public class C extends B
{
public static void main(String args[])
{
A Ob_A1 = new A();
B Ob_B1 = new B();
Ob_A1.A1();
System.out.println();
Ob_B1.B1();
}
}
1280 × 800px
200%
EN
10:42 PM
4/30/2012

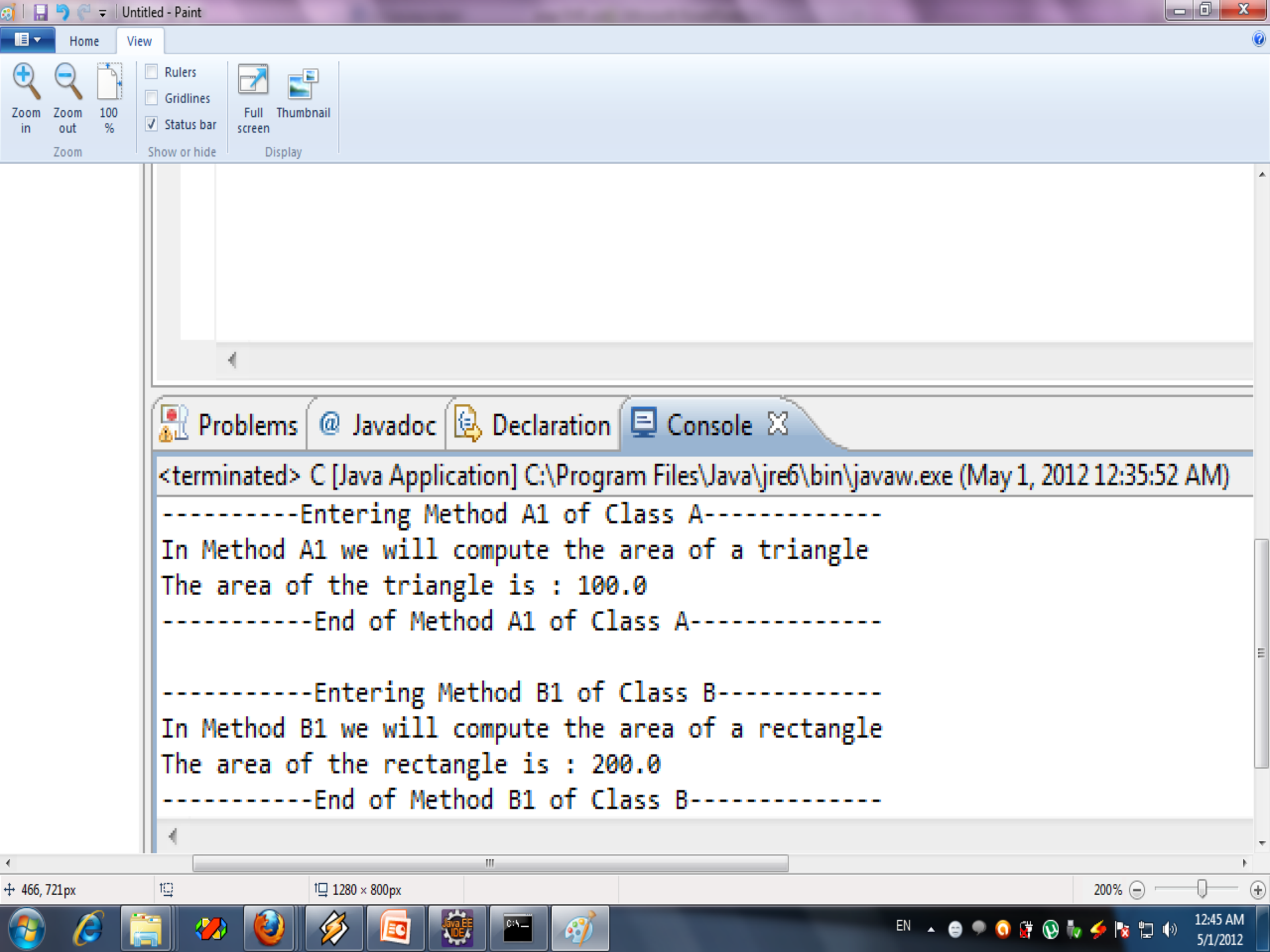

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> C [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (Apr 30, 2012 10:41:08 PM)
----------Entering Method A1 of Class A----------
In Method A1 we will compute the area of a triangle
The area of the triangle is : 100.0
----------End of Method A1 of Class A----------
----------Entering Method B1 of Class B----------
In Method B1 we will compute the area of a rectangle
The area of the rectangle is : 200.0
----------End of Method B1 of Class B----------
405, 640px
1280 × 800px
200%
EN
10:45 PM
4/30/2012

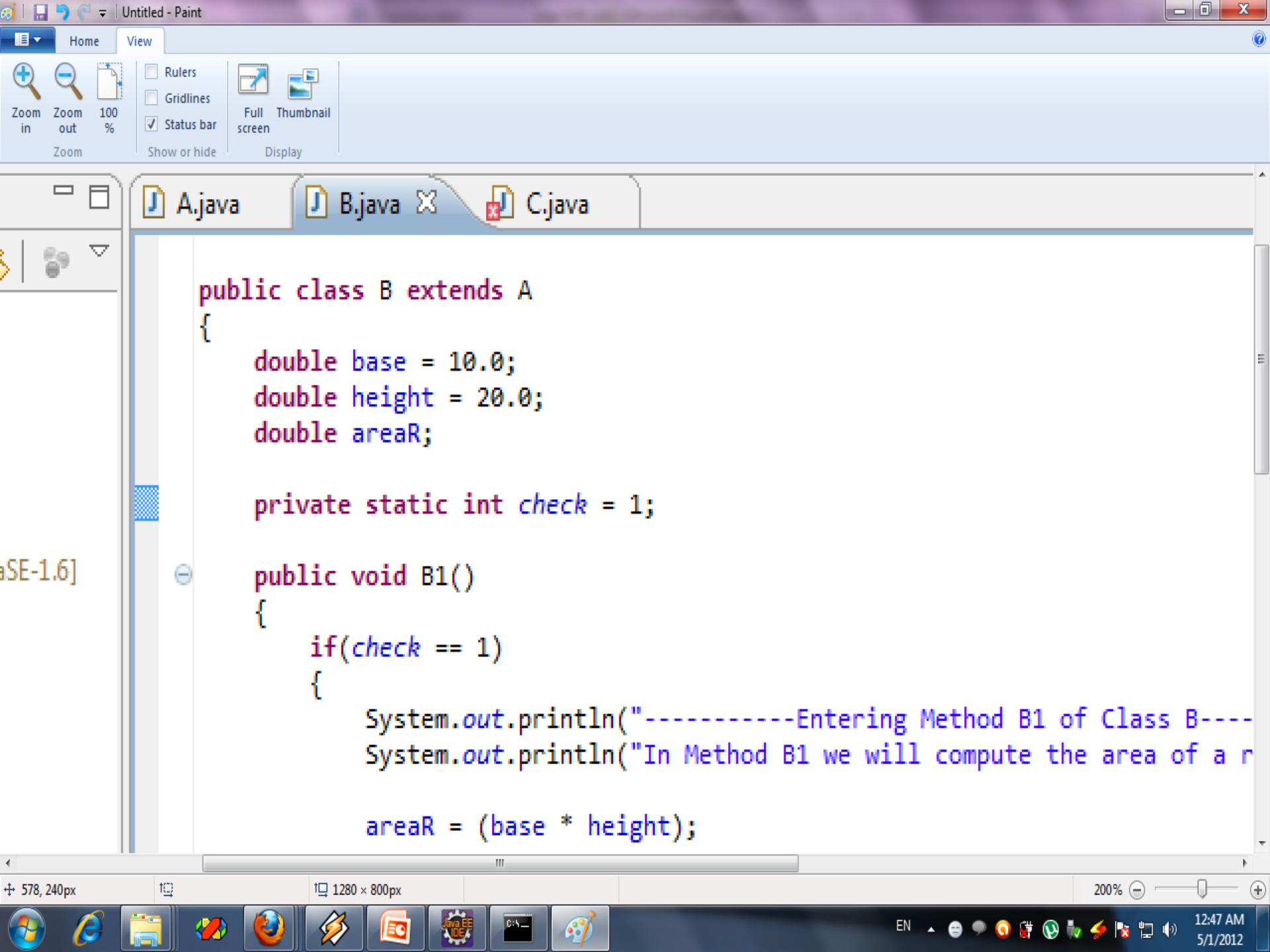
Inheritance

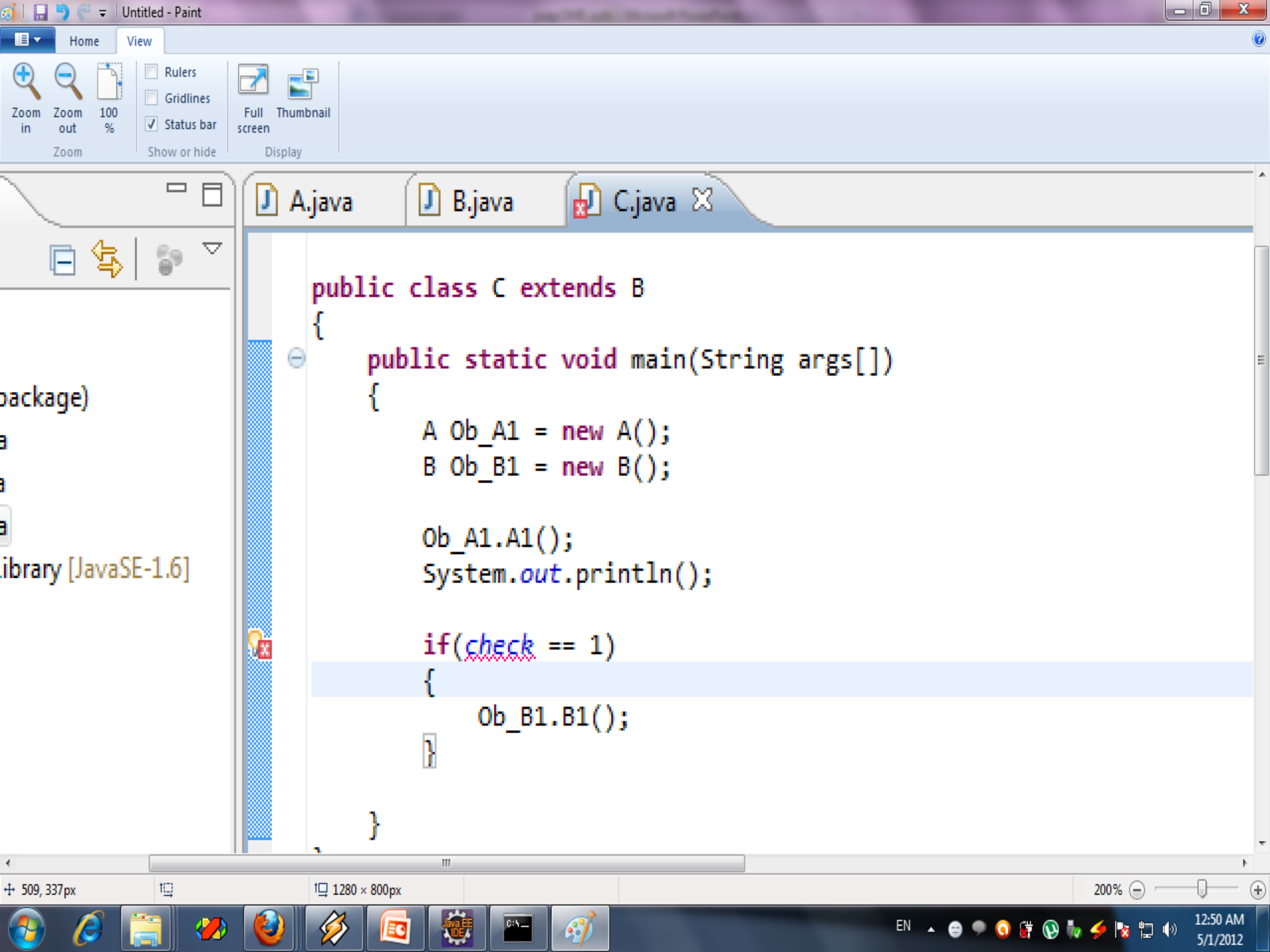
- An important note to remember while doing work with inheritance:
  1. Although a subclass includes all the members of its superclass in one level hierarchy and also includes all the members of the superclass of which its superclass is a subclass. (following is an example)
  2. One subclass cannot have multiple superclass, but a superclass can have multiple subclass. (already shown in the first example)

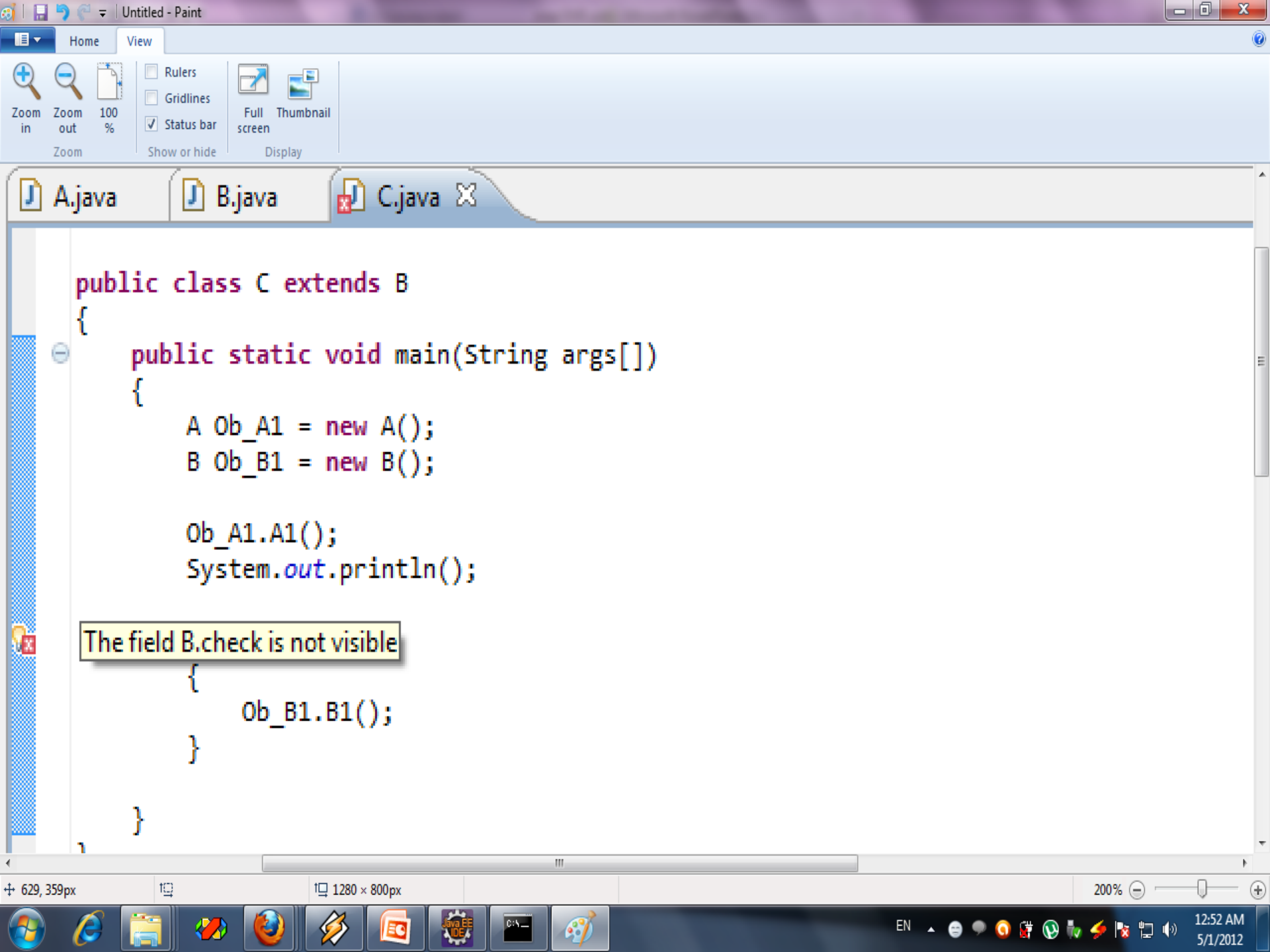
```
public class B extends A
{
    double base = 10.0;
    double height = 20.0;
    double areaR;

    public static int check = 1;

    public void B1()
    {
        if(check == 1)
        {
            System.out.println("----------Entering Method B1 of Class B----------");
            System.out.println("In Method B1 we will compute the area of a rectangle");

            areaR = (base * height);

            System.out.println("The area of the rectangle is : " + areaR);
            System.out.println("----------End of Method B1 of Class B----------");
        }
```

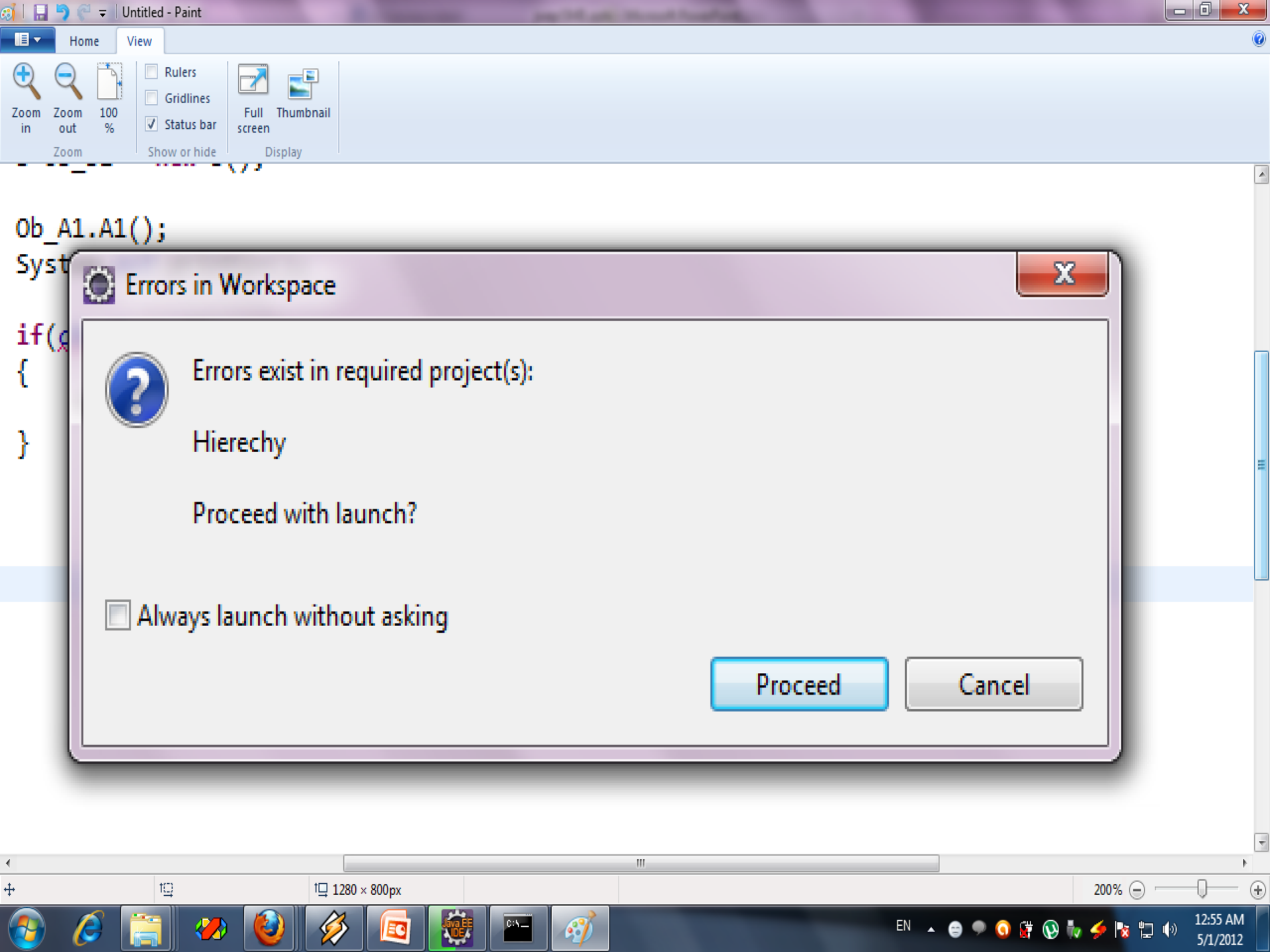

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Package Explorer
Hierechy
src
(default package)
A.java
B.java
C.java
JRE System Library [JavaSE-1.6]
Inherit
A.java
B.java
C.java
public class C extends B
{
public static void main(String args[])
{
A Ob_A1 = new A();
B Ob_B1 = new B();
Ob_A1.A1();
System.out.println();
if(check == 1)
{
Ob_B1.B1();
}
}
312, 255px
1280 × 800px
200%
EN
12:44 AM
5/1/2012

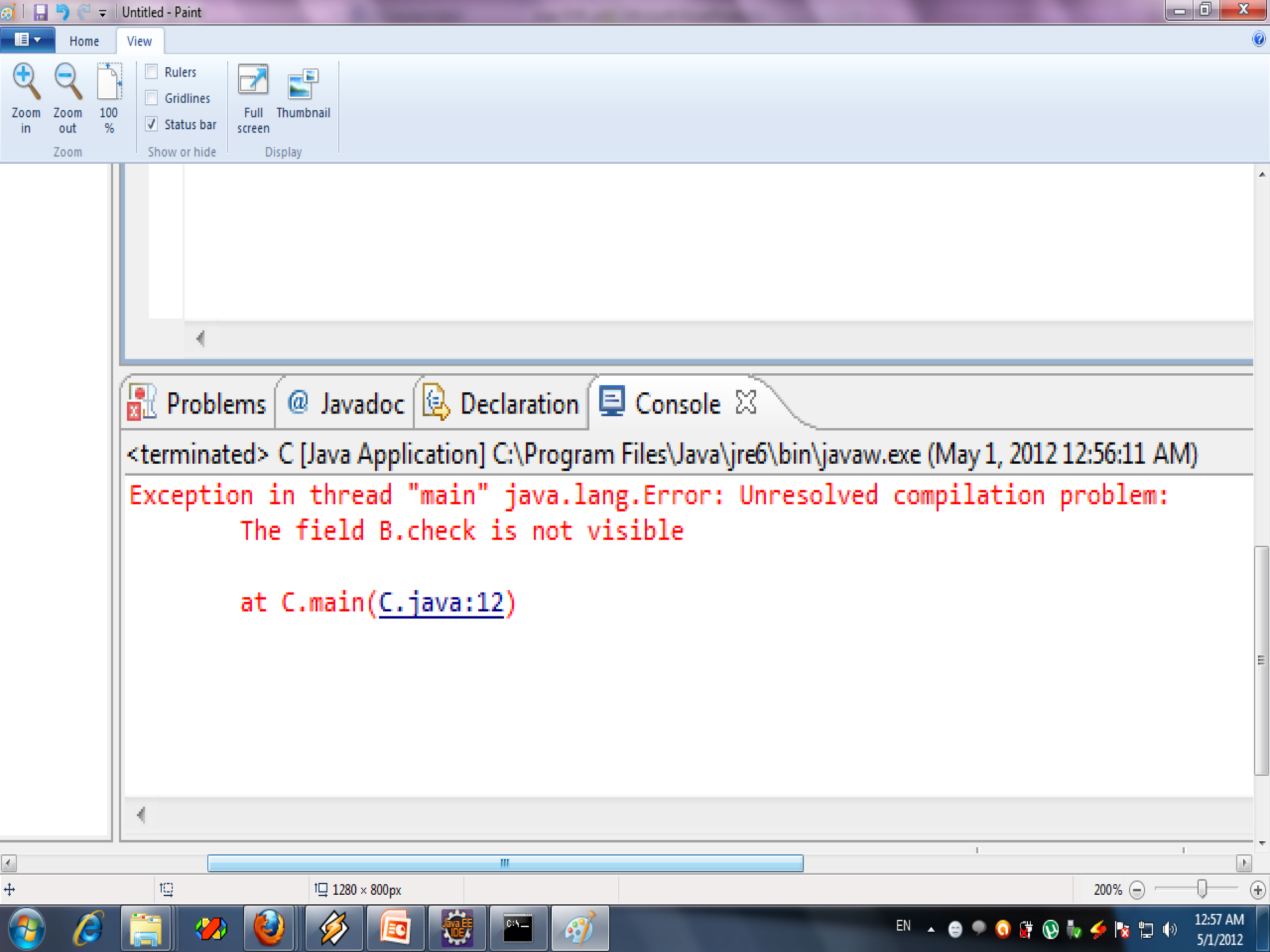

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> C [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 1, 2012 12:35:52 AM)
----------Entering Method A1 of Class A----------
In Method A1 we will compute the area of a triangle
The area of the triangle is : 100.0
----------End of Method A1 of Class A----------
----------Entering Method B1 of Class B----------
In Method B1 we will compute the area of a rectangle
The area of the rectangle is : 200.0
----------End of Method B1 of Class B----------
466, 721px
1280 × 800px
200%
EN
12:45 AM
5/1/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
A.java
B.java
C.java
public class B extends A
{
double base = 10.0;
double height = 20.0;
double areaR;
private static int check = 1;
public void B1()
{
if(check == 1)
{
System.out.println("----------Entering Method B1 of Class B----
System.out.println("In Method B1 we will compute the area of a r
areaR = (base * height);
578, 240px
1280 × 800px
200%
EN
12:47 AM
5/1/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
A.java
B.java
C.java
public class C extends B
{
public static void main(String args[])
{
A Ob_A1 = new A();
B Ob_B1 = new B();
Ob_A1.A1();
System.out.println();
if(check == 1)
{
Ob_B1.B1();
}
}
package)
ibrary [JavaSE-1.6]
509, 337px
1280 × 800px
200%
EN
12:50 AM
5/1/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
A.java
B.java
C.java
public class C extends B
{
public static void main(String args[])
{
A Ob_A1 = new A();
B Ob_B1 = new B();
Ob_A1.A1();
System.out.println();
The field B.check is not visible
{
Ob_B1.B1();
}
}
}
629, 359px
1280 × 800px
200%
EN
12:52 AM
5/1/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Ob_A1.A1();
Errors in Workspace
Errors exist in required project(s):
Hierechy
Proceed with launch?
Always launch without asking
Proceed
Cancel
1280 × 800px
200%
EN
12:55 AM
5/1/2012

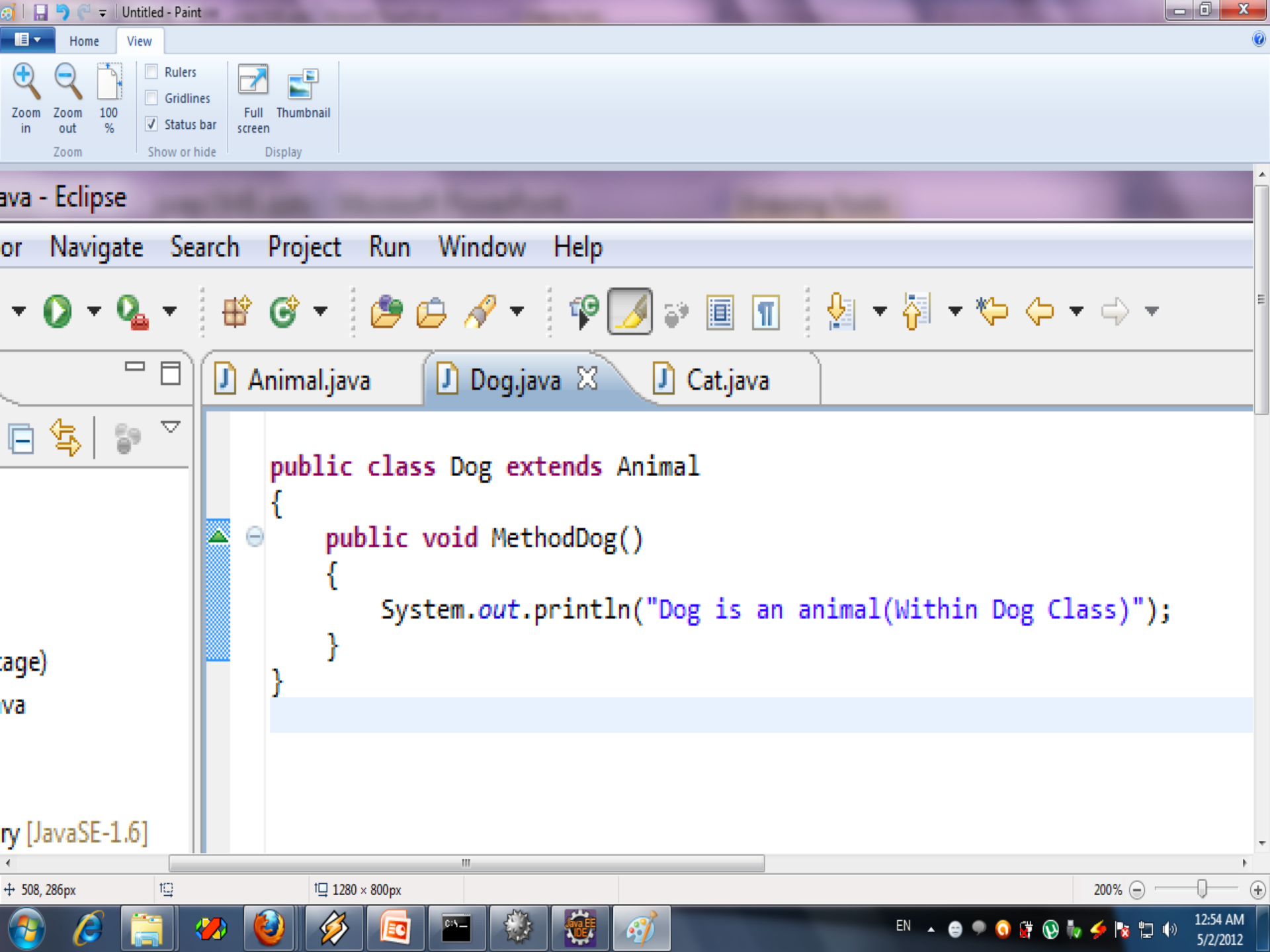

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> C [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 1, 2012 12:56:11 AM)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The field B.check is not visible
at C.main(C.java:12)
1280 × 800px
200%
EN
12:57 AM
5/1/2012

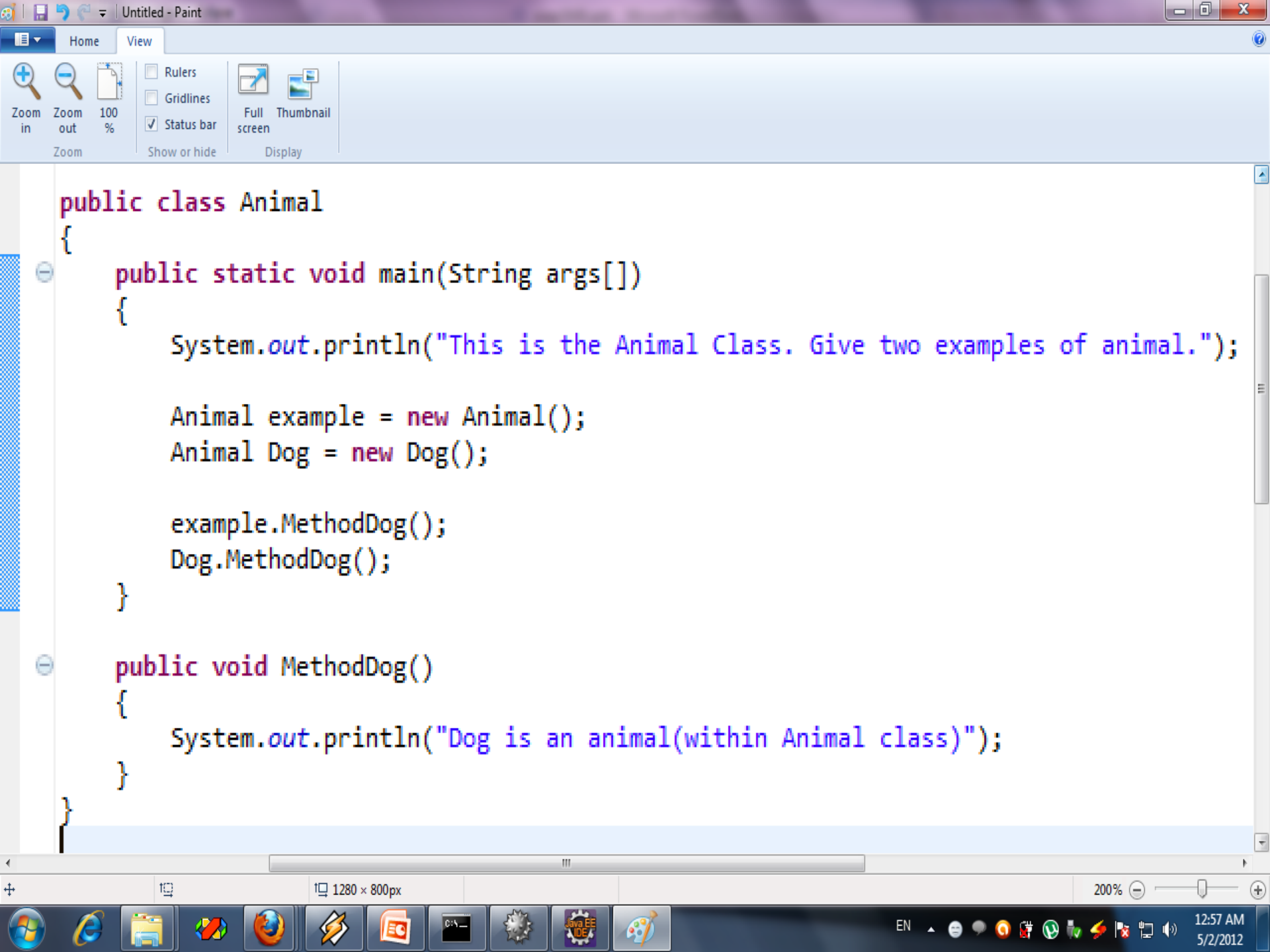
Method Overriding

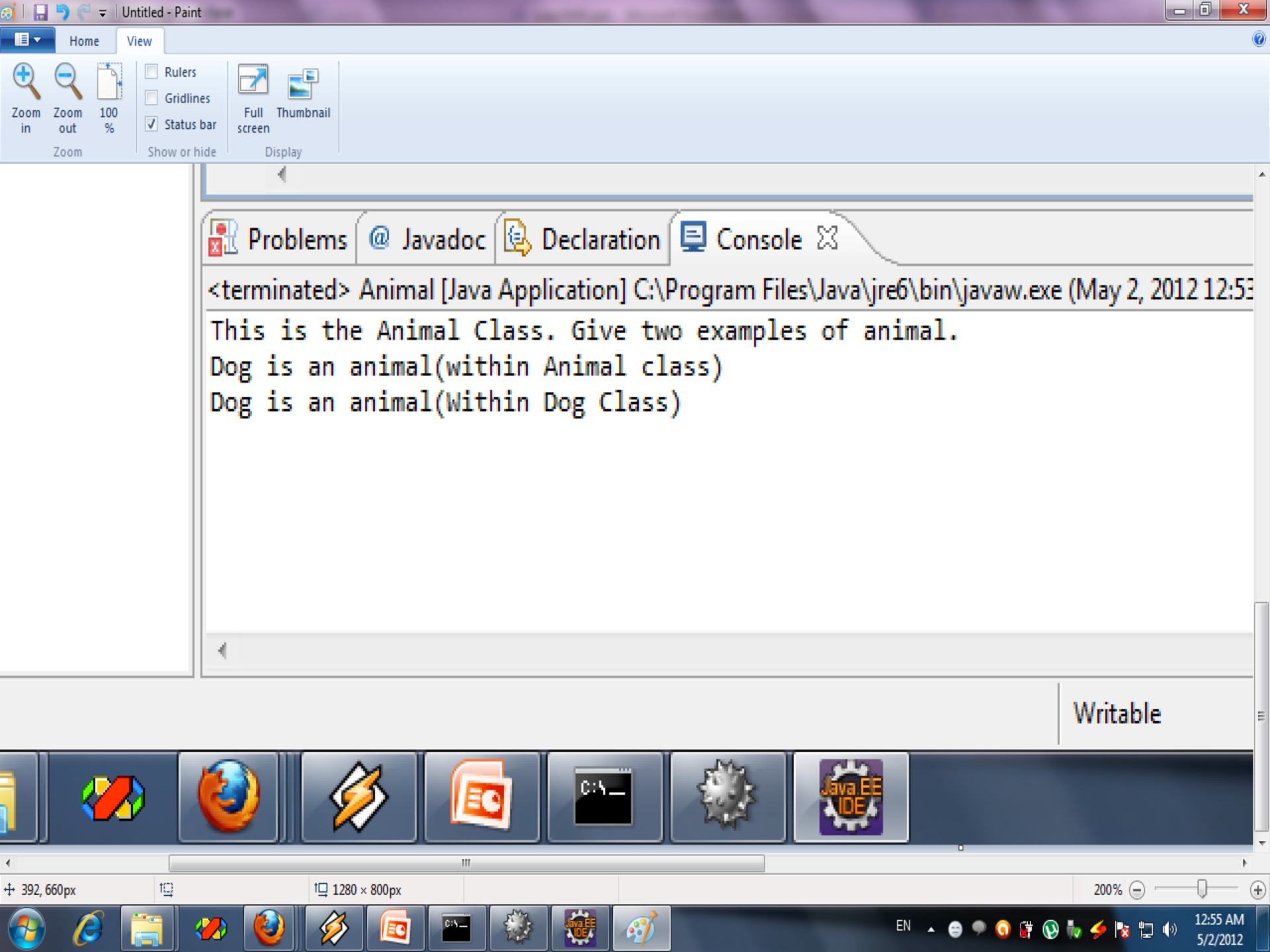
- In a class hierarchy, when a method in a subclass has the same name and type signature as a method in its superclass,
  - Then ***the method in the subclass is said to override the method in the superclass.***
- When *an overridden method is called from within a subclass*,
  - It will always ***refer to the version of that method defined by the subclass***.
  - The version of the method defined by the superclass will be ***hidden***.

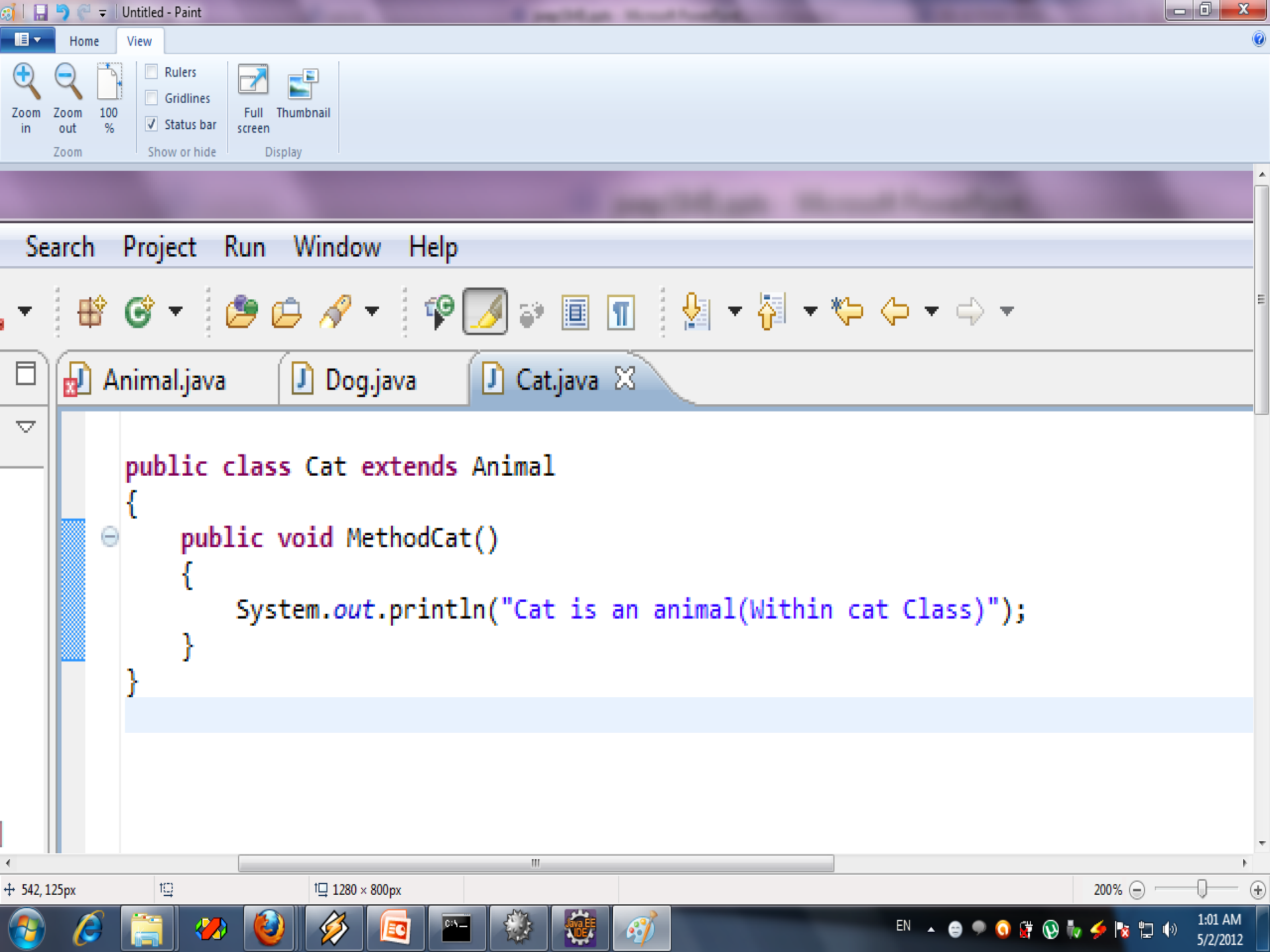
Method Overriding

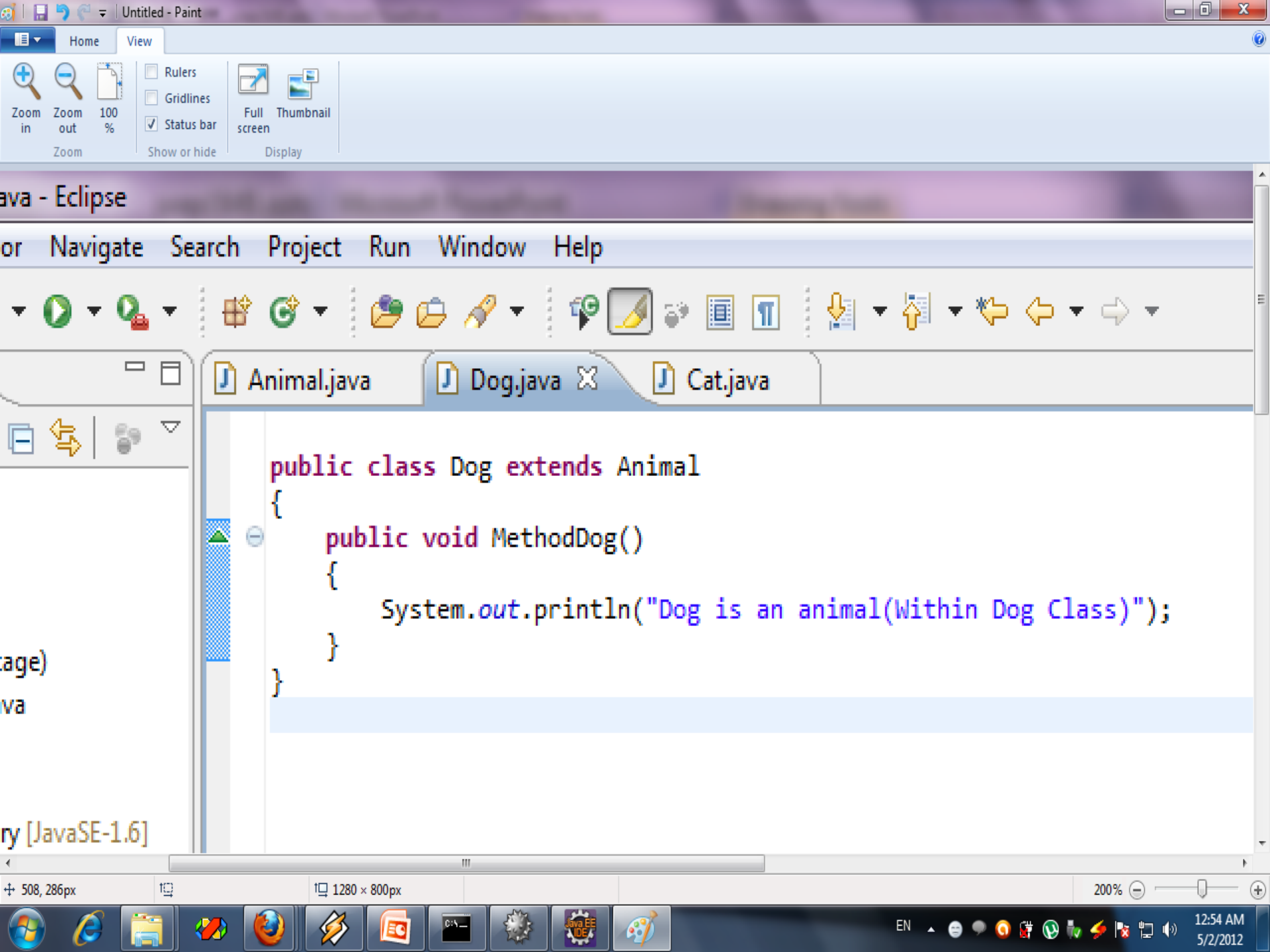
- Following are two examples :
  1. Shows that how a method in a subclass with same signature that of superclass overrides the method of superclass.
  2. When overriding gives error.

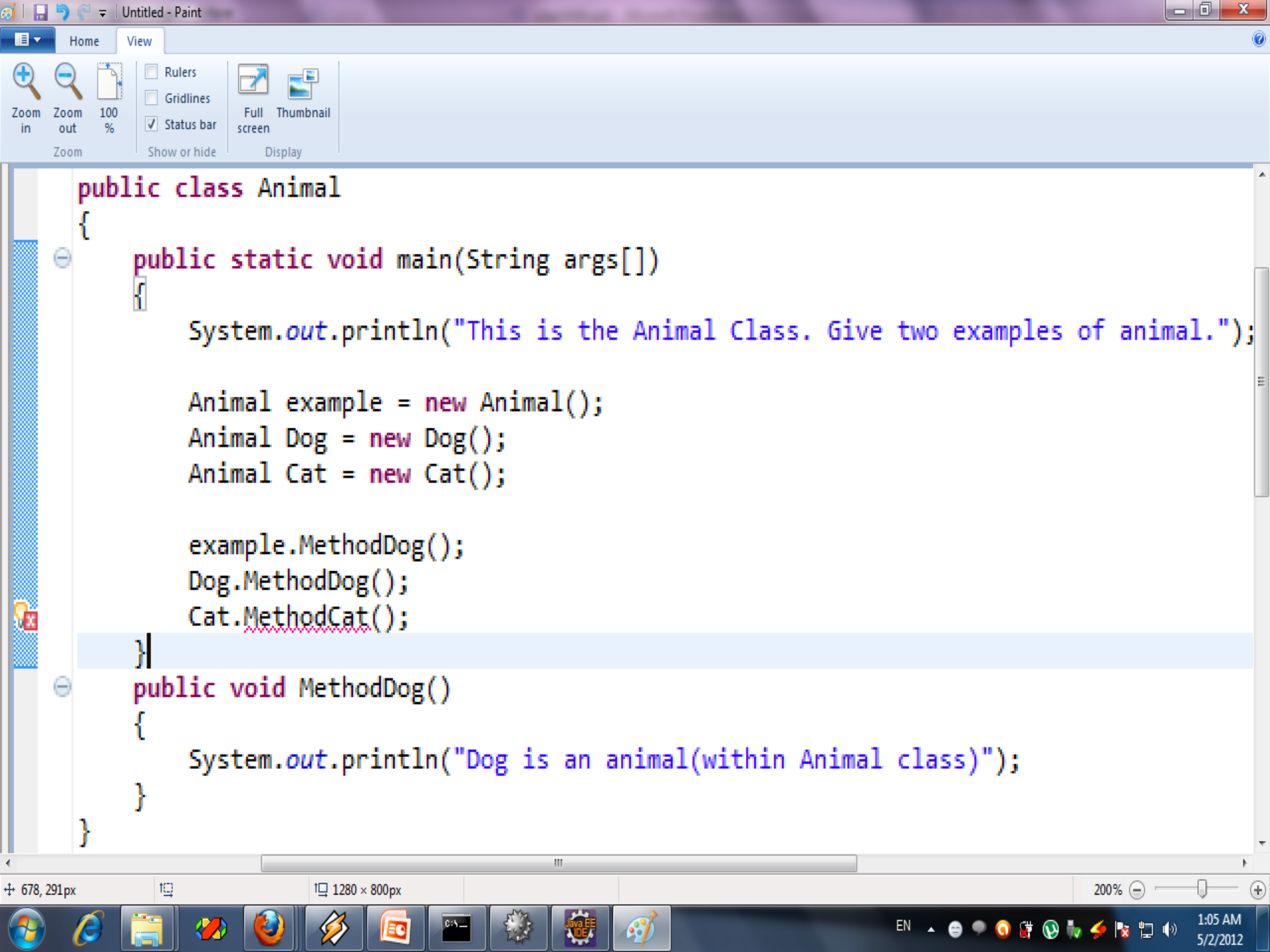

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
ava - Eclipse
or Navigate Search Project Run Window Help
Animal.java
Dog.java
Cat.java
public class Dog extends Animal
{
public void MethodDog()
{
System.out.println("Dog is an animal(Within Dog Class)");
}
}
cage)
va
ry [JavaSE-1.6]
508, 286px
1280 × 800px
200%
EN
12:54 AM
5/2/2012

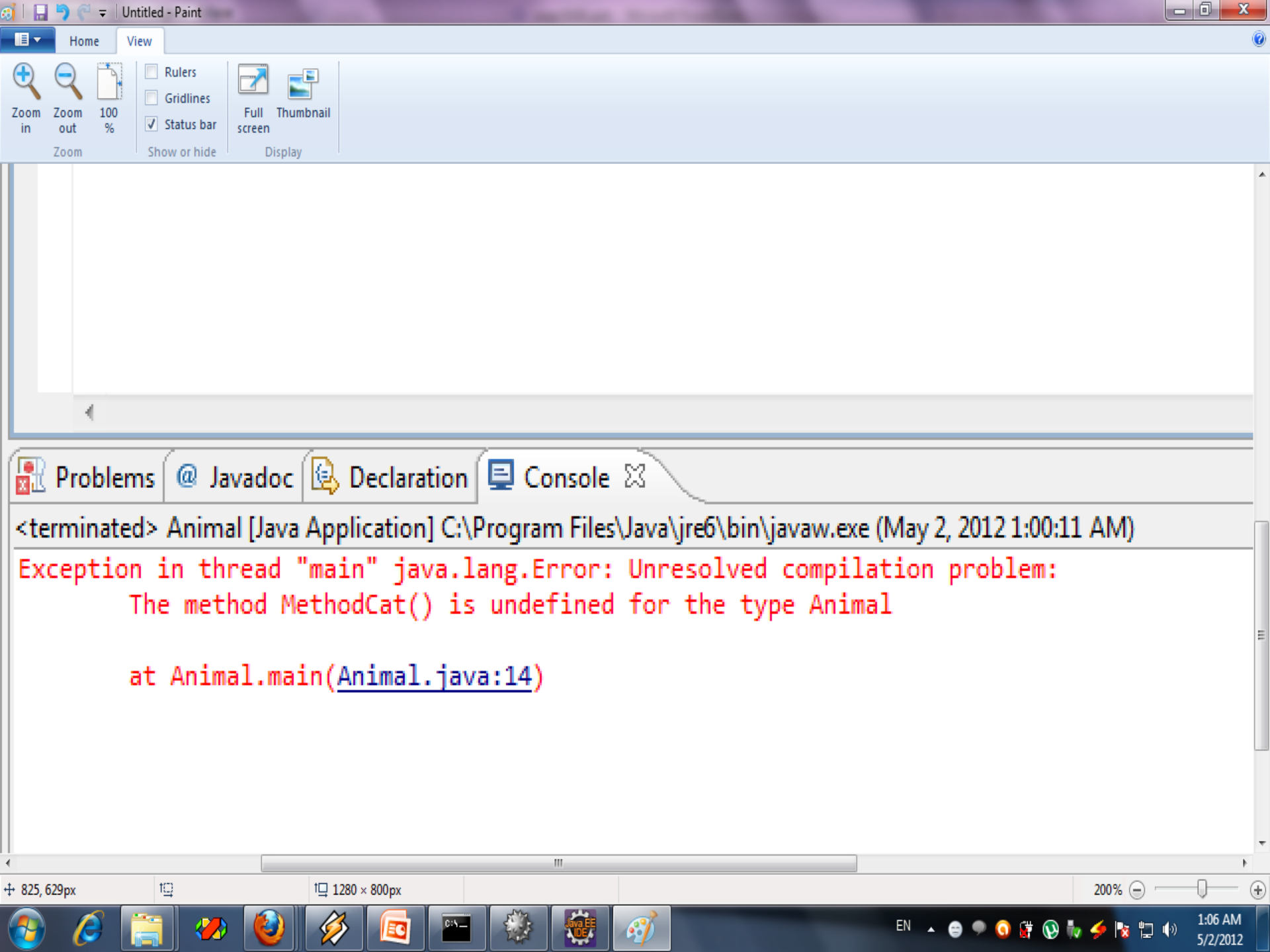

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
public class Animal
{
public static void main(String args[])
{
System.out.println("This is the Animal Class. Give two examples of animal.");
Animal example = new Animal();
Animal Dog = new Dog();
example.MethodDog();
Dog.MethodDog();
}
public void MethodDog()
{
System.out.println("Dog is an animal(within Animal class)");
}
}
1280 × 800px
200%
EN
12:57 AM
5/2/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> Animal [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 2, 2012 12:53
This is the Animal Class. Give two examples of animal.
Dog is an animal(within Animal class)
Dog is an animal(Within Dog Class)
Writable
392, 660px
1280 × 800px
200%
EN
12:55 AM
5/2/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Search
Project
Run
Window
Help
Animal.java
Dog.java
Cat.java
public class Cat extends Animal
{
public void MethodCat()
{
System.out.println("Cat is an animal(Within cat Class)");
}
}
542, 125px
1280 × 800px
200%
EN
1:01 AM
5/2/2012

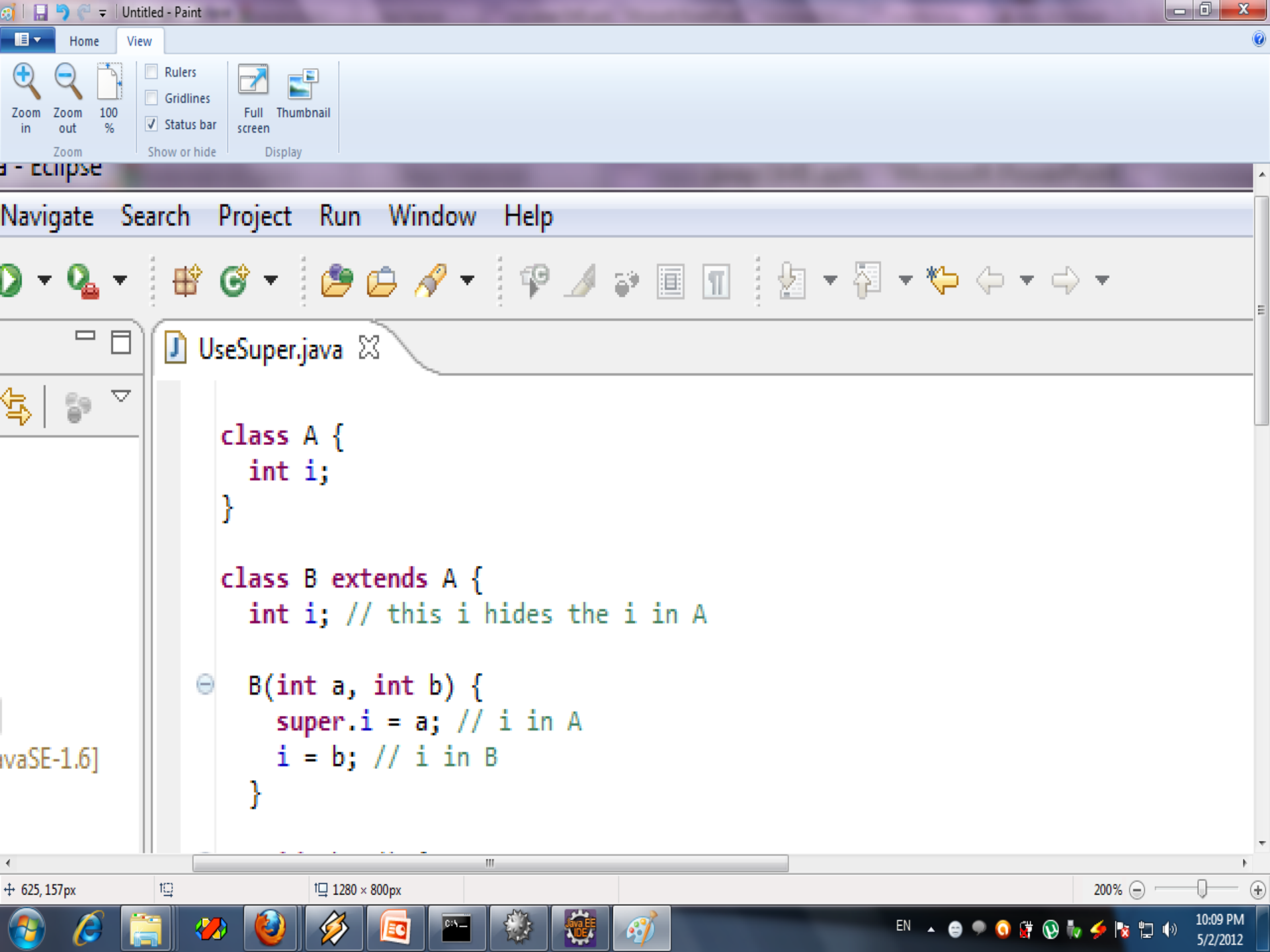

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
ava - Eclipse
or Navigate Search Project Run Window Help
Animal.java
Dog.java
Cat.java
public class Dog extends Animal
{
public void MethodDog()
{
System.out.println("Dog is an animal(Within Dog Class)");
}
}
cage)
va
ry [JavaSE-1.6]
508, 286px
1280 × 800px
200%
EN
12:54 AM
5/2/2012

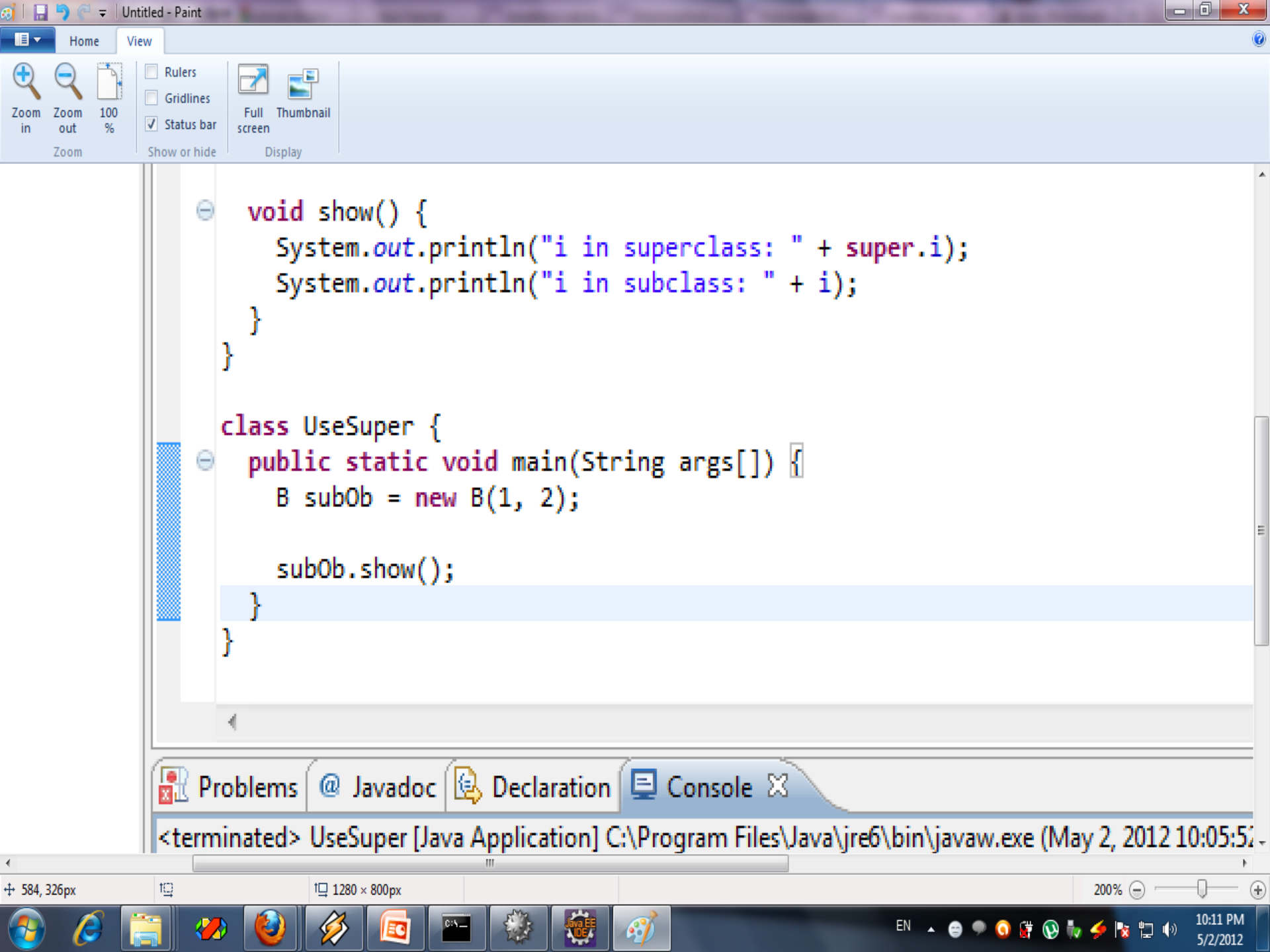

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
public class Animal
{
public static void main(String args[])
{
System.out.println("This is the Animal Class. Give two examples of animal.");
Animal example = new Animal();
Animal Dog = new Dog();
Animal Cat = new Cat();
example.MethodDog();
Dog.MethodDog();
Cat.MethodCat();
}
public void MethodDog()
{
System.out.println("Dog is an animal(within Animal class)");
}
}
678, 291px
1280 × 800px
200%
EN
1:05 AM
5/2/2012

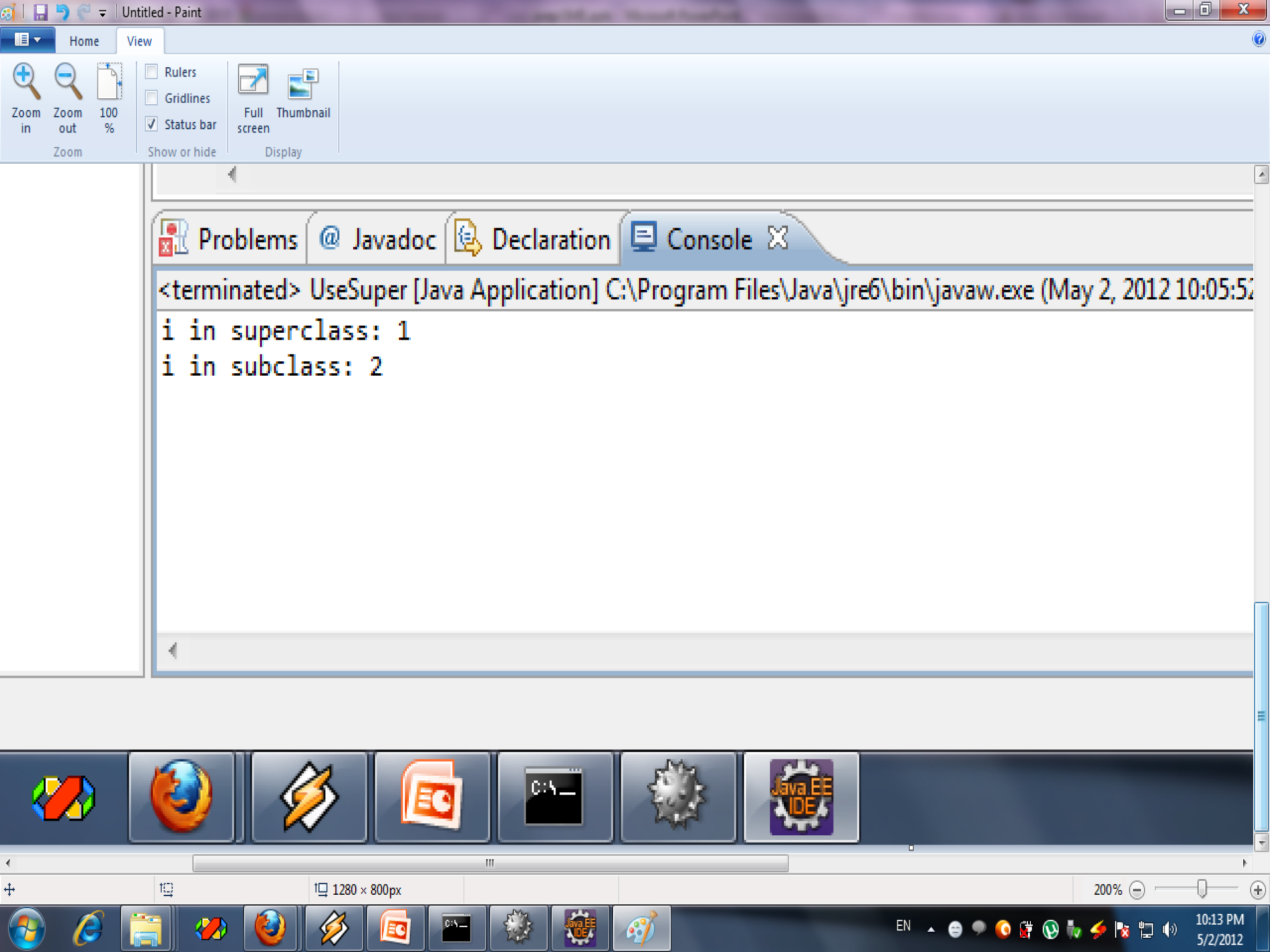

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> Animal [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 2, 2012 1:00:11 AM)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The method MethodCat() is undefined for the type Animal
at Animal.main(Animal.java:14)
825, 629px
1280 × 800px
200%
EN
1:06 AM
5/2/2012

Using Super

- Whenever a subclass needs to refer to its immediate superclass, it can do so by use of the ***keyword super***.
- Super has two general forms:
  1. The first ***calls the superclass' constructor***.
  2. The second is ***used to access a member of the superclass*** that has been hidden by a member of a subclass.
- Following is an example of using super:


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Navigate
Search
Project
Run
Window
Help
UseSuper.java
class A {
int i;
}
class B extends A {
int i; // this i hides the i in A
B(int a, int b) {
super.i = a; // i in A
i = b; // i in B
}
625, 157px
1280 × 800px
200%
EN
10:09 PM
5/2/2012

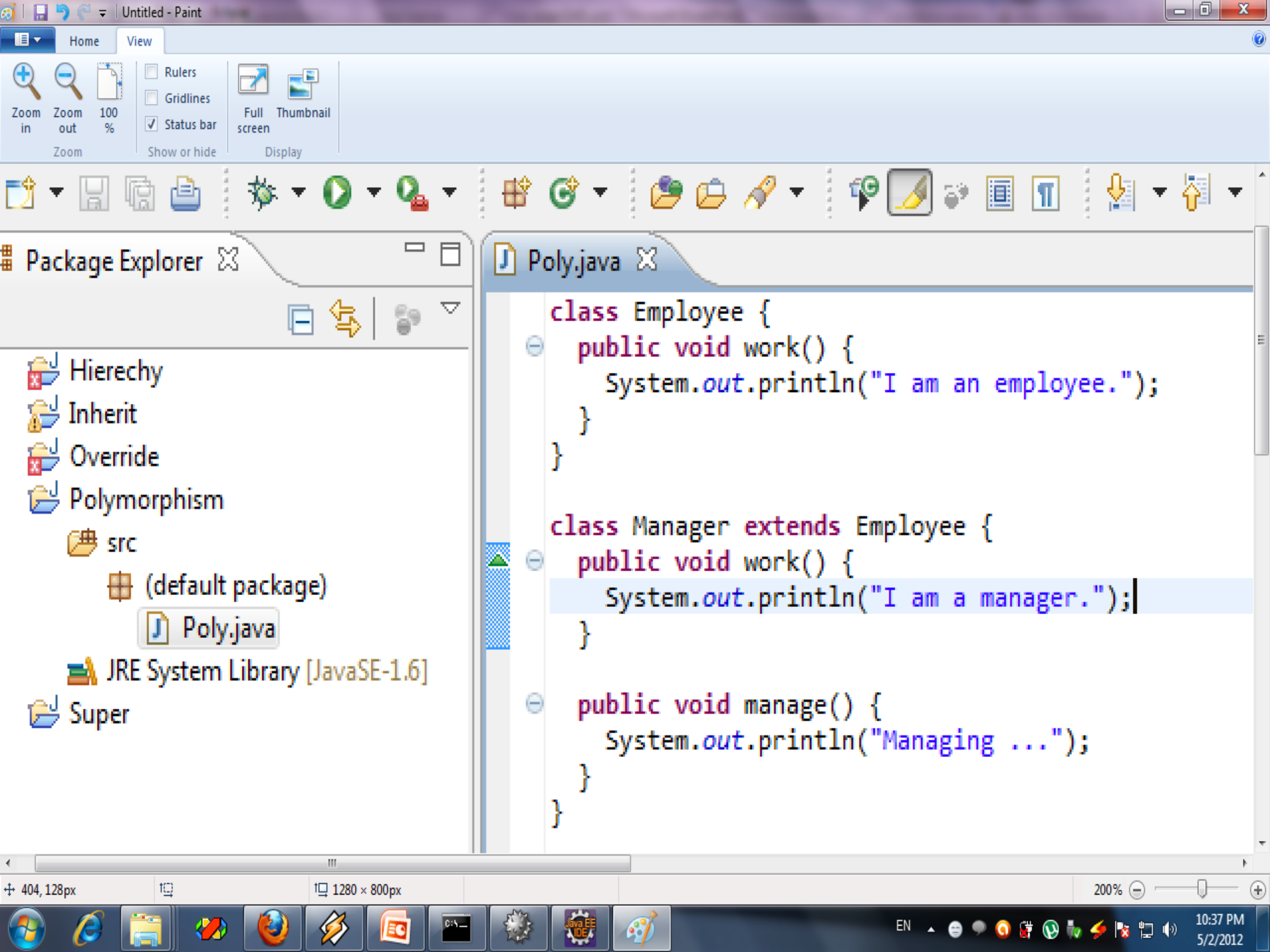

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
void show() {
System.out.println("i in superclass: " + super.i);
System.out.println("i in subclass: " + i);
}
}
class UseSuper {
public static void main(String args[]) {
B subOb = new B(1, 2);
subOb.show();
}
}
Problems
Javadoc
Declaration
Console
<terminated> UseSuper [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 2, 2012 10:05:5
584, 326px
1280 × 800px
200%
EN
10:11 PM
5/2/2012

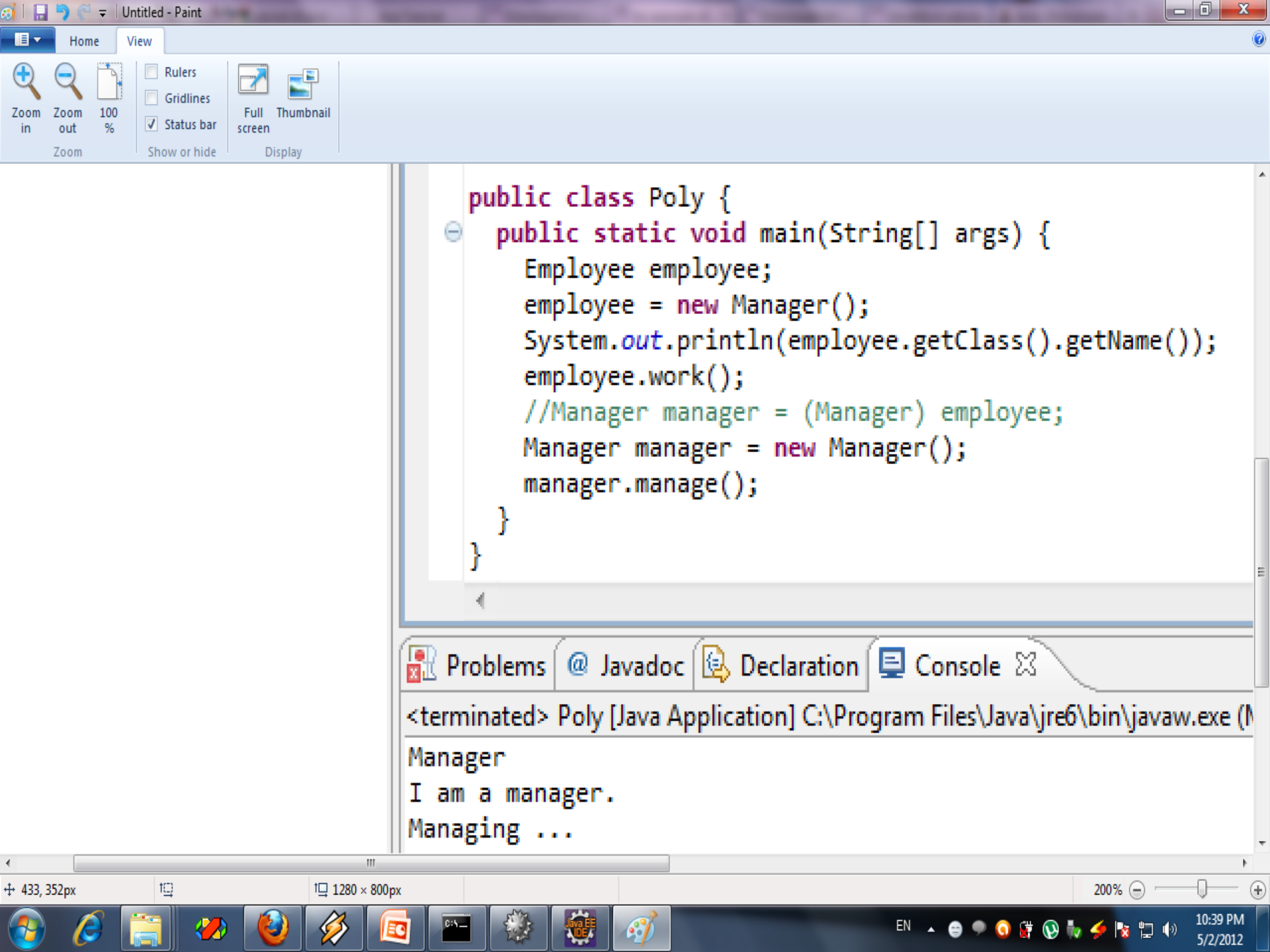

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> UseSuper [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 2, 2012 10:05:52
i in superclass: 1
i in subclass: 2
1280 × 800px
200%
EN
10:13 PM
5/2/2012

Polymorphism

Polymorphism

- Polymorphism is the ***ability of an object to take on many forms***.
- The ***most common use*** of polymorphism in OOP occurs when ***a parent class reference is used to refer to a child class object***.
- It means the ability of ***a single variable of a given type*** to be used ***to reference objects of different types and to automatically call the method that is specific to the type*** of object the variable references.


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Package Explorer
Hierechy
Inherit
Override
Polymorphism
src
(default package)
Poly.java
JRE System Library [JavaSE-1.6]
Super
Poly.java
class Employee {
public void work() {
System.out.println("I am an employee.");
}
}
class Manager extends Employee {
public void work() {
System.out.println("I am a manager.");
}
public void manage() {
System.out.println("Managing ...");
}
}
404, 128px
1280 × 800px
200%
EN
10:37 PM
5/2/2012

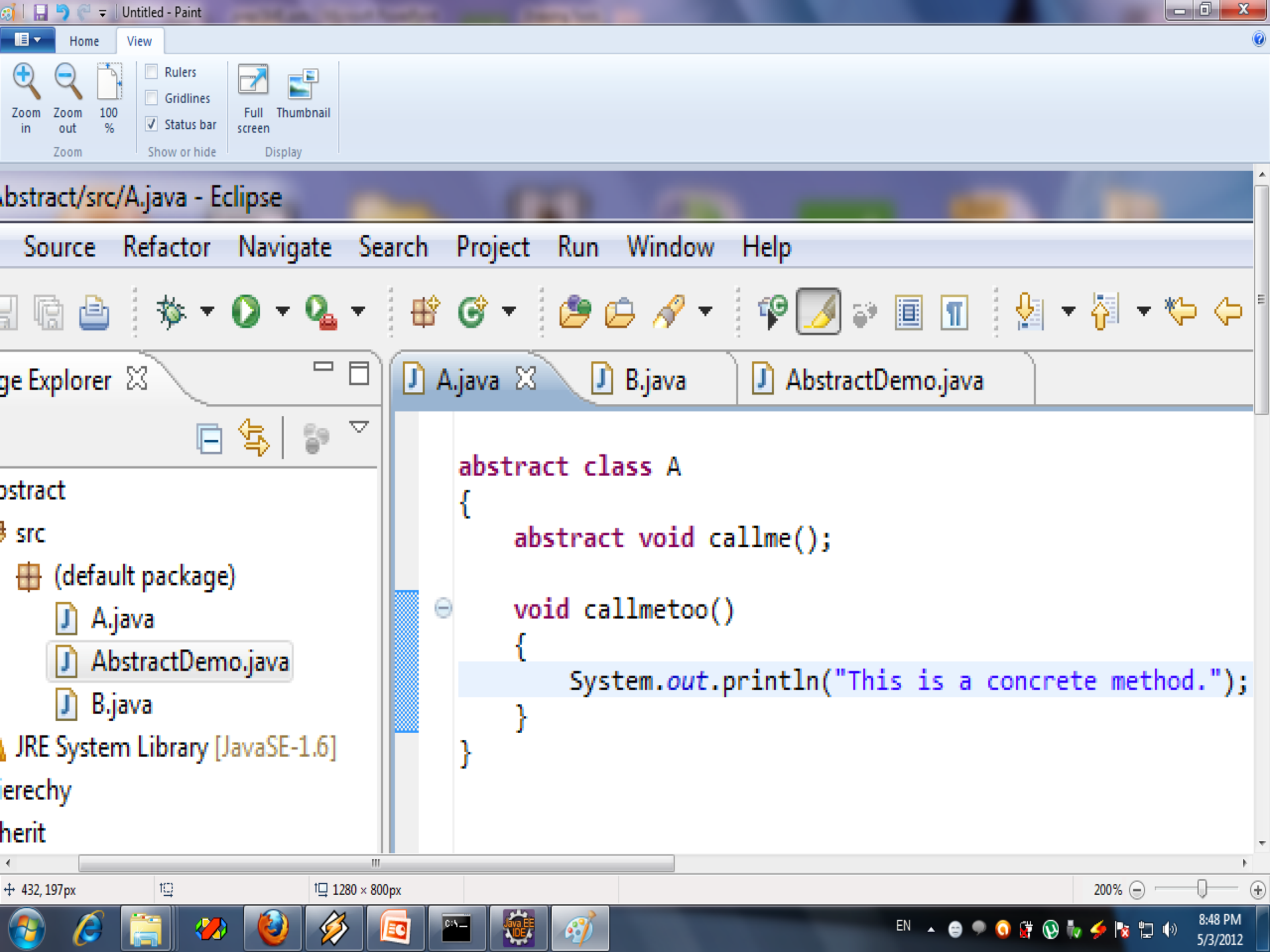

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
public class Poly {
public static void main(String[] args) {
Employee employee;
employee = new Manager();
System.out.println(employee.getClass().getName());
employee.work();
//Manager manager = (Manager) employee;
Manager manager = new Manager();
manager.manage();
}
}
Problems
Javadoc
Declaration
Console
<terminated> Poly [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (
Manager
I am a manager.
Managing ...
433, 352px
1280 × 800px
200%
EN
10:39 PM
5/2/2012

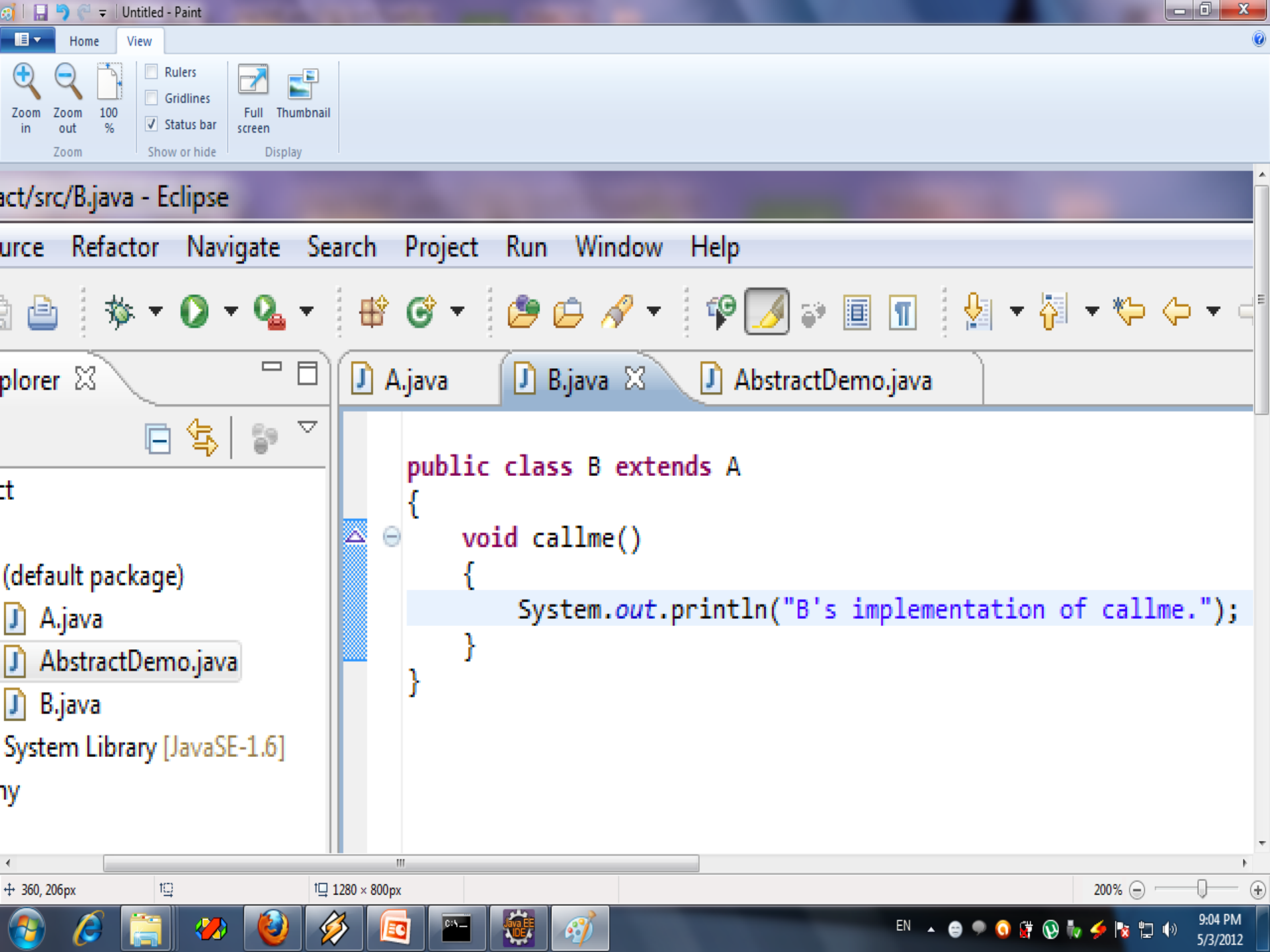
Using Abstract Classes

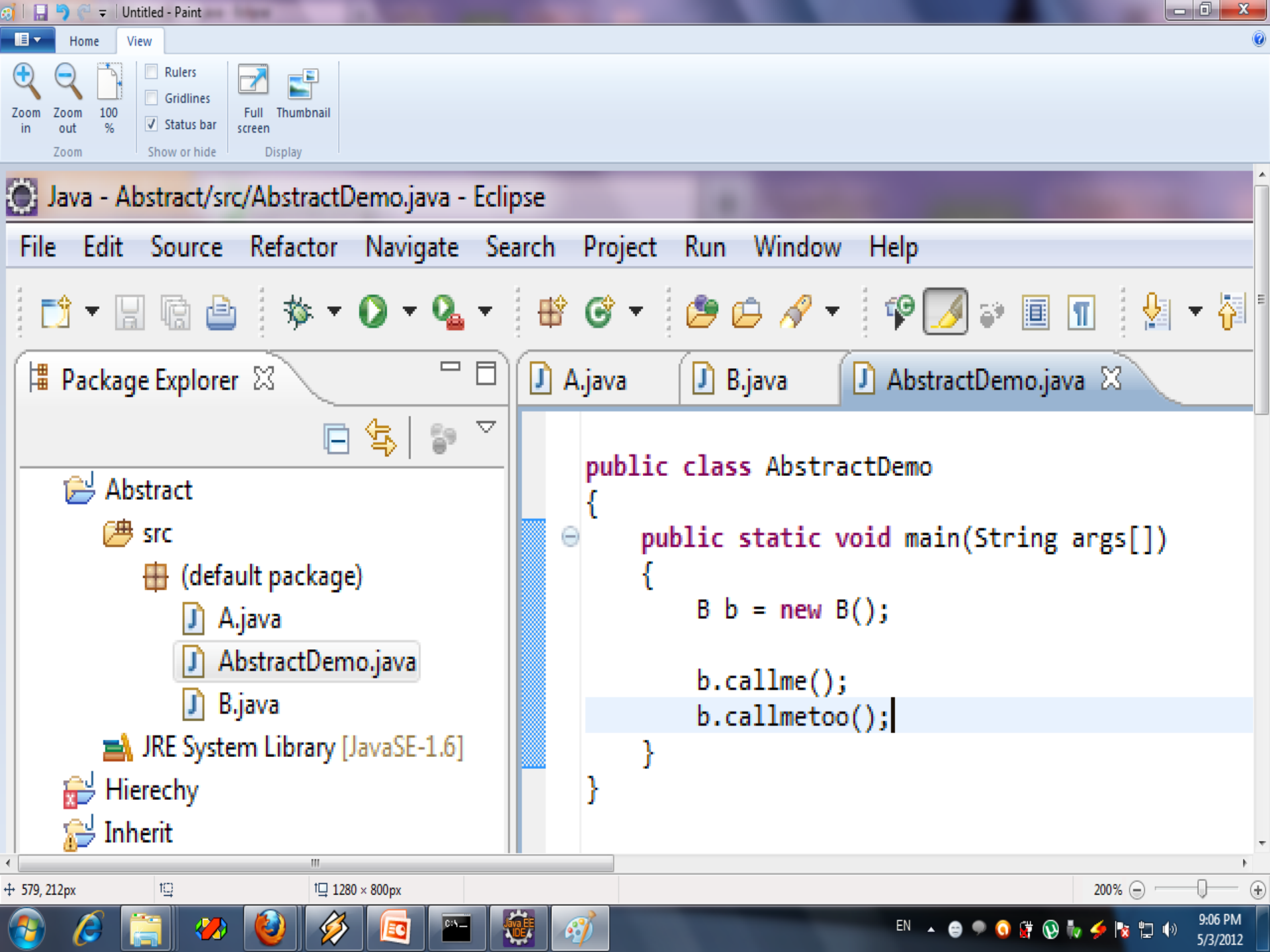
- Sometimes it is required that certain methods be overridden by subclasses by specifying the ***abstract*** type.
- These methods are sometimes called ***subclasser responsibility*** because they have no implementation specified in the superclass.
- ***To declare a class abstract***, you simple ***use the abstract keyword in front of the class*** keyword at the beginning of the class declaration.

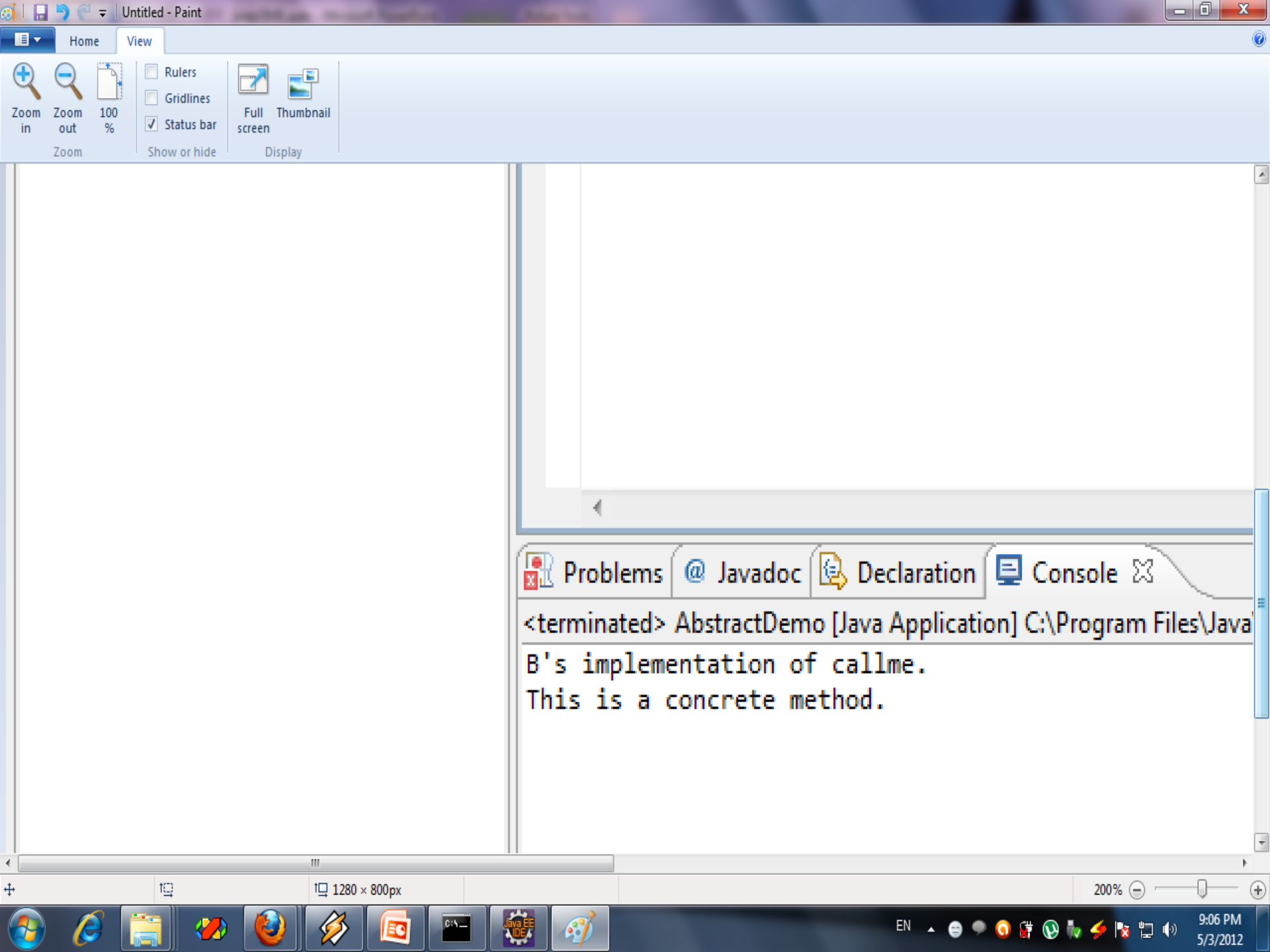
Using Abstract Classes

- There can be ***no objects*** of an abstract class.
- An abstract class ***cannot be directly instantiated*** with the new operator.
- Such ***objects would be useless***, because an ***abstract class is not fully defined.***
- Since class of superclass is overridden, so its implementation is not necessary.(advantage)
- Following is an example :


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
bstract/src/A.java - Eclipse
Source
Refactor
Navigate
Search
Project
Run
Window
Help
ge Explorer
A.java
B.java
AbstractDemo.java
abstract class A
{
abstract void callme();
void callmetoo()
{
System.out.println("This is a concrete method.");
}
}
bstract
src
(default package)
A.java
AbstractDemo.java
B.java
JRE System Library [JavaSE-1.6]
erechy
herit
432, 197px
1280 × 800px
200%
EN
8:48 PM
5/3/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
act/src/B.java - Eclipse
urce Refactor Navigate Search Project Run Window Help
plorer
A.java
B.java
AbstractDemo.java
public class B extends A
{
void callme()
{
System.out.println("B's implementation of callme.");
}
}
ct
(default package)
A.java
AbstractDemo.java
B.java
System Library [JavaSE-1.6]
ny
360, 206px
1280 × 800px
200%
EN
9:04 PM
5/3/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Java - Abstract/src/AbstractDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Abstract
src
(default package)
A.java
AbstractDemo.java
B.java
JRE System Library [JavaSE-1.6]
Hierechy
Inherit
A.java
B.java
AbstractDemo.java
public class AbstractDemo
{
public static void main(String args[])
{
B b = new B();
b.callme();
b.callmetoo();
}
}
579, 212px
1280 × 800px
200%
EN
9:06 PM
5/3/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> AbstractDemo [Java Application] C:\Program Files\Java
B's implementation of callme.
This is a concrete method.
1280 × 800px
200%
EN
9:06 PM
5/3/2012

Using Final

- Using final to prevent overriding:
  - While method overriding in one of the JAVA's most powerful features, there will be times when we do not want overriding.
  - We can do this by using ***final*** keyword.
  - With final we can ***prevent a method to be overridden*** as well as we can ***prevent a class to be extended, that is prevent Inheritance***.

Using final to prevent Overriding

```
Class A
{
      final void meth()
      {
            System.out.println("This is a final method.");
      }
}

Class B extends A
{
      void meth()
      {
            //ERROR! Can't override the "meth" method
            System.out.println("Illegal !!!");
       }
}
```

Using final to prevent Inheritance

```
final class A
{
      void meth()
      {
            System.out.println("Method in final class.");
      }
}

//The following class is illegal.
Class B extends A
//can not be a subclass of A which is final
{


}
```

Exception Handling

Exception Handling Fundamentals

- A JAVA exception is an object that describes an exception(or, an error) condition that has occurred in a block of a code.
- When an exceptional condition arises, an object representing that exception is created and ***thrown*** in the method that caused the error.
- At some point the exception is ***caught*** and processed.

Exception Handling Fundamentals

- Java exception handling is managed via five keywords:
  - Try, catch, throw, throws, and finally
- Program statements that you want to monitor for exceptions are contained within a ***try*** block.
- If an exception occurs within the ***try*** block, it is thrown

Exception Handling Fundamentals

- Your code can ***catch*** this exception and can handle it.
- System generated exceptions are automatically thrown by the java run-time system.
- To manually throw an exception, use the keyword ***throw***.
-

Exception Handling Fundamentals

- Any exception that is thrown out of a method must be specified as such by a ***throws*** clause.
- Any code that absolutely must be executed before a method returns is put in a ***finally*** block.

Exception Types

- All exception types are subclasses of the built in class ***Throwable***.
- There are two subclasses that partitions exceptions into two distinct branches.
  1. Exception
     - This class is used for exceptional conditions that user program should catch.
     - From this class, you can create your own custom exception types.

Exception Types

- There is an important subclass of Exception, called ***RuntimeException***.
- Runtime exceptions are automatically defined.

2. Error
   - Defines exceptions that are not expected to be caught under normal circumstances by your program.
   - This is not handled by your program.
   - This type of error can crash your program.

Uncaught Exception

- Following is a program segment within which the “Divide by Zero” error is intentionally occurred.

```
class Exc0
{
     public static void main(String args[])
     {
          int d = 0;
          int a = 42 / d;
     }
}
```

Uncaught Exception

- When the JAVA run time system detects the attempt to ***divide by zero***, it constructs a new exception object and then *throws* the exception.
- This causes the execution of Exc0 to stop, now it must be caught by an exception handler.
- In this example, we have not supplied any exception handler of our own.

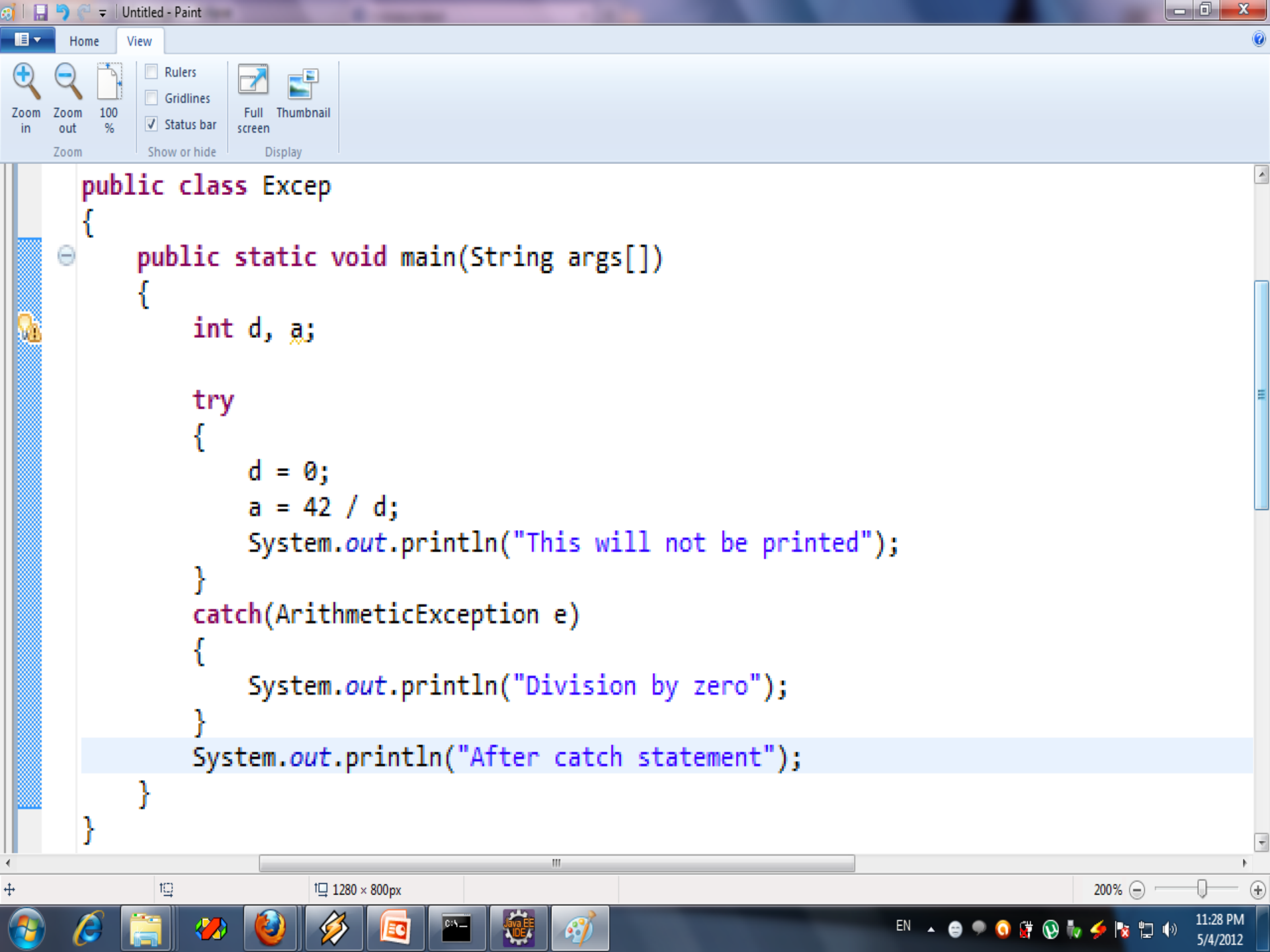
Uncaught Exception

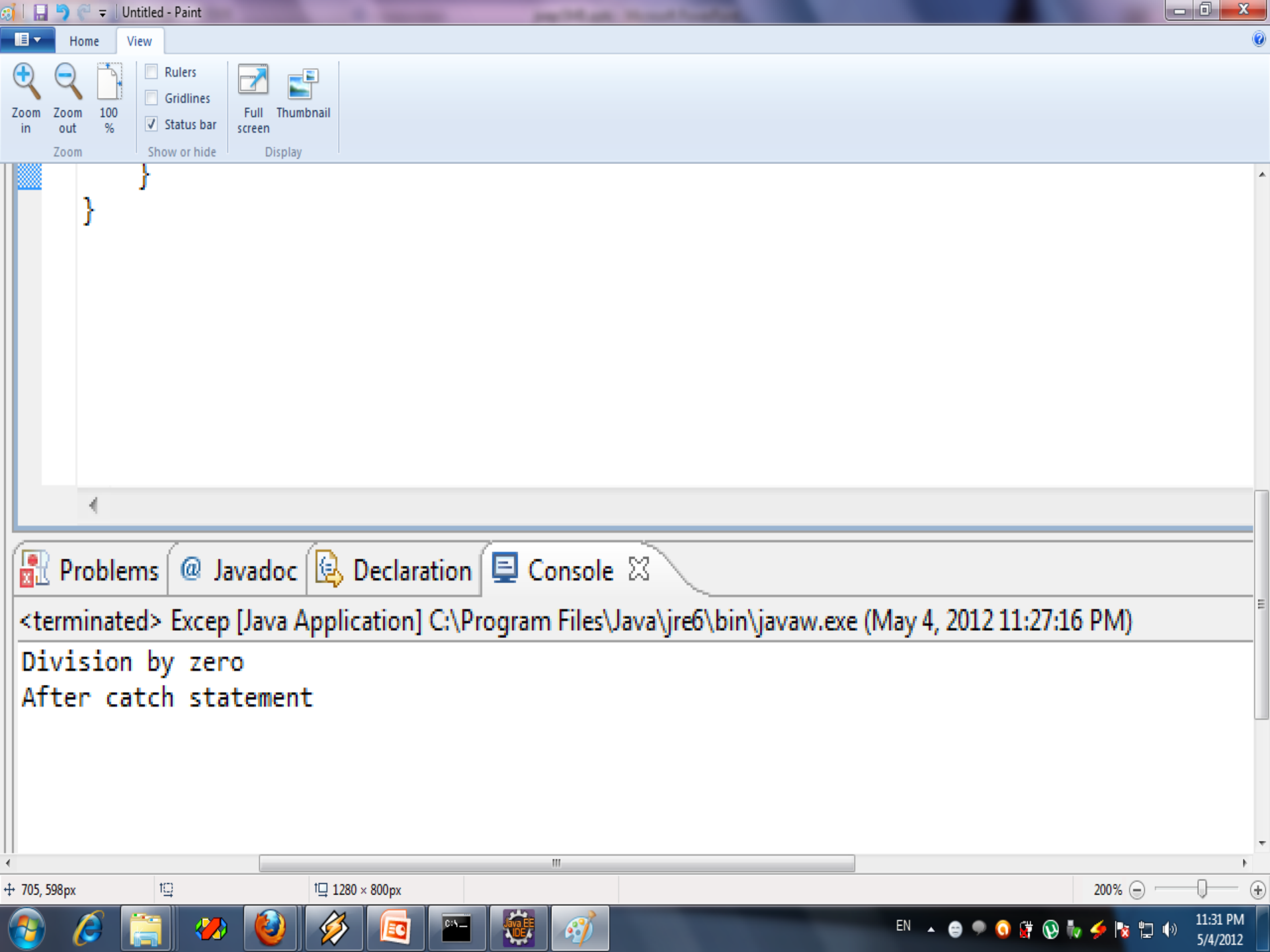
- So it is caught by the default handler provided by the java run time system.
- The default handler
  - displays a string describing the exception
  - Prints the point at which the exception occurred
  - And, terminates the program.
- For previous exception results:

```
Java.lang.ArithmeticException: / by zero
                    at Exc0.main(Exc0.java:4)
```

Using Try-Catch()

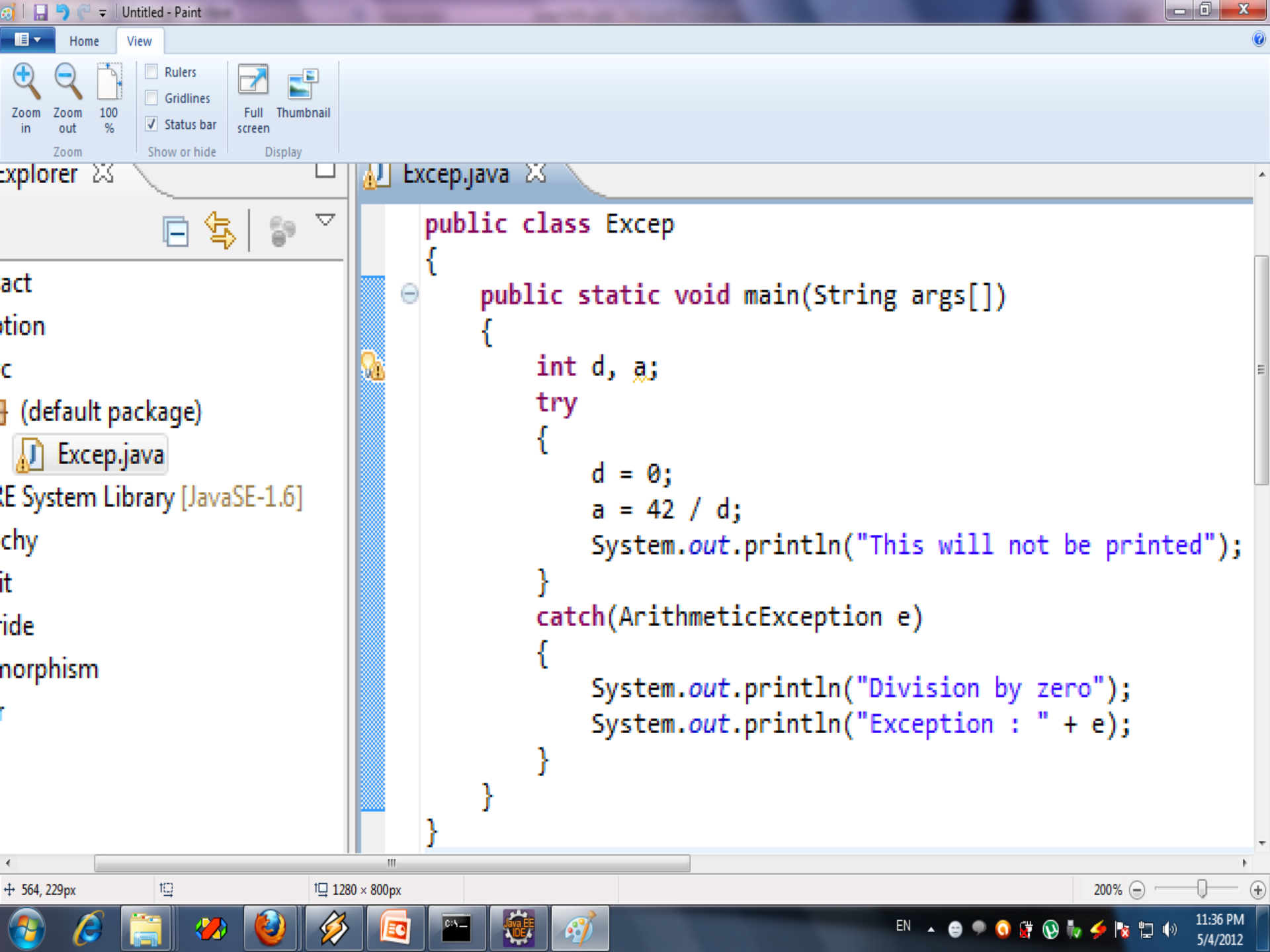
- Now the examples
  1. ***using try-catch and our own handler***
  2. Using try-catch and printing default string corresponding to the exception.
  3. Multiple catch clauses
  4. Nested try statement

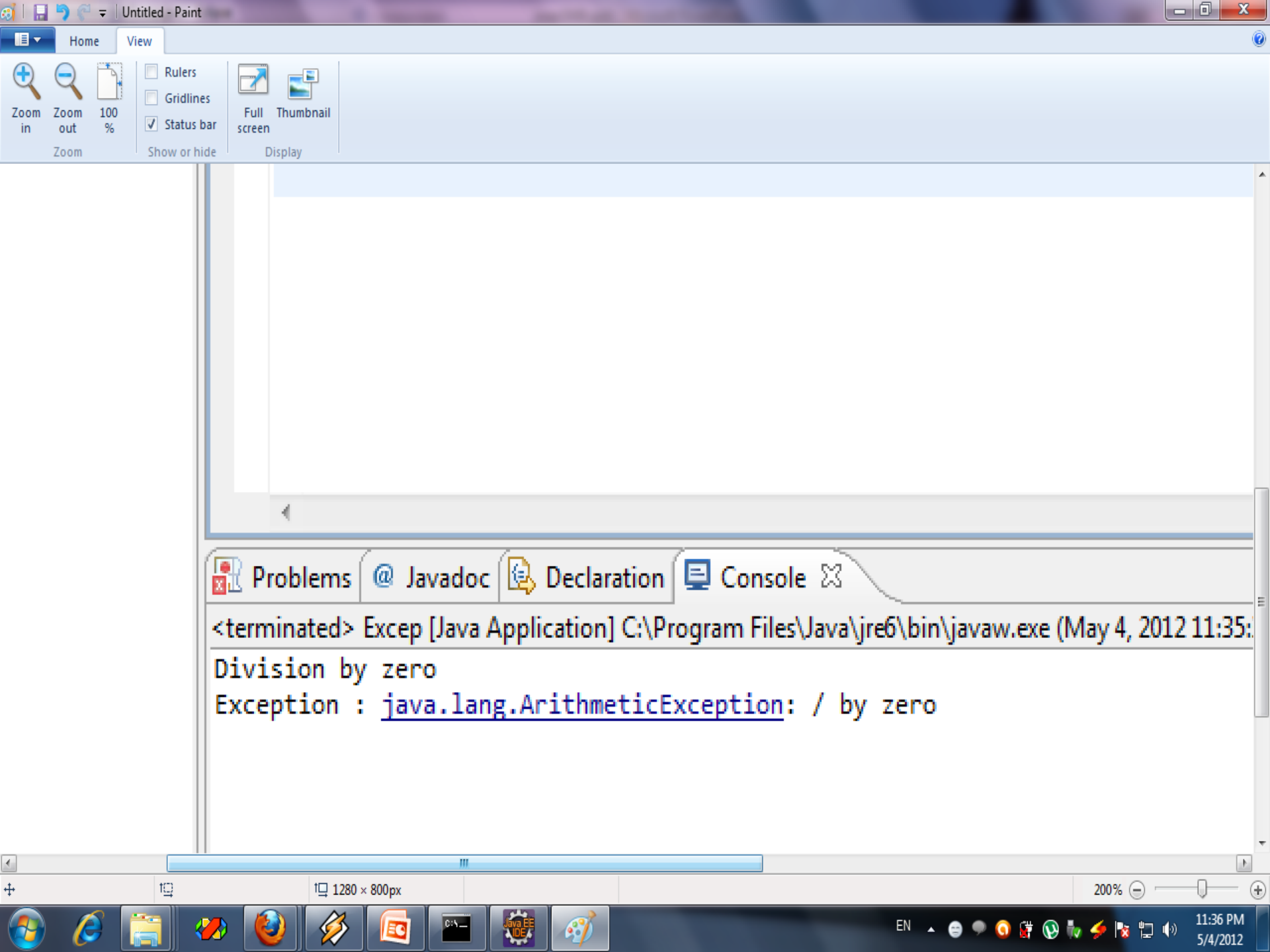

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
public class Excep
{
public static void main(String args[])
{
int d, a;
try
{
d = 0;
a = 42 / d;
System.out.println("This will not be printed");
}
catch(ArithmeticException e)
{
System.out.println("Division by zero");
}
System.out.println("After catch statement");
}
}
1280 × 800px
200%
EN
11:28 PM
5/4/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
}
}
Problems
Javadoc
Declaration
Console
<terminated> Excep [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 4, 2012 11:27:16 PM)
Division by zero
After catch statement
705, 598px
1280 × 800px
200%
EN
11:31 PM
5/4/2012

Using Try-Catch()

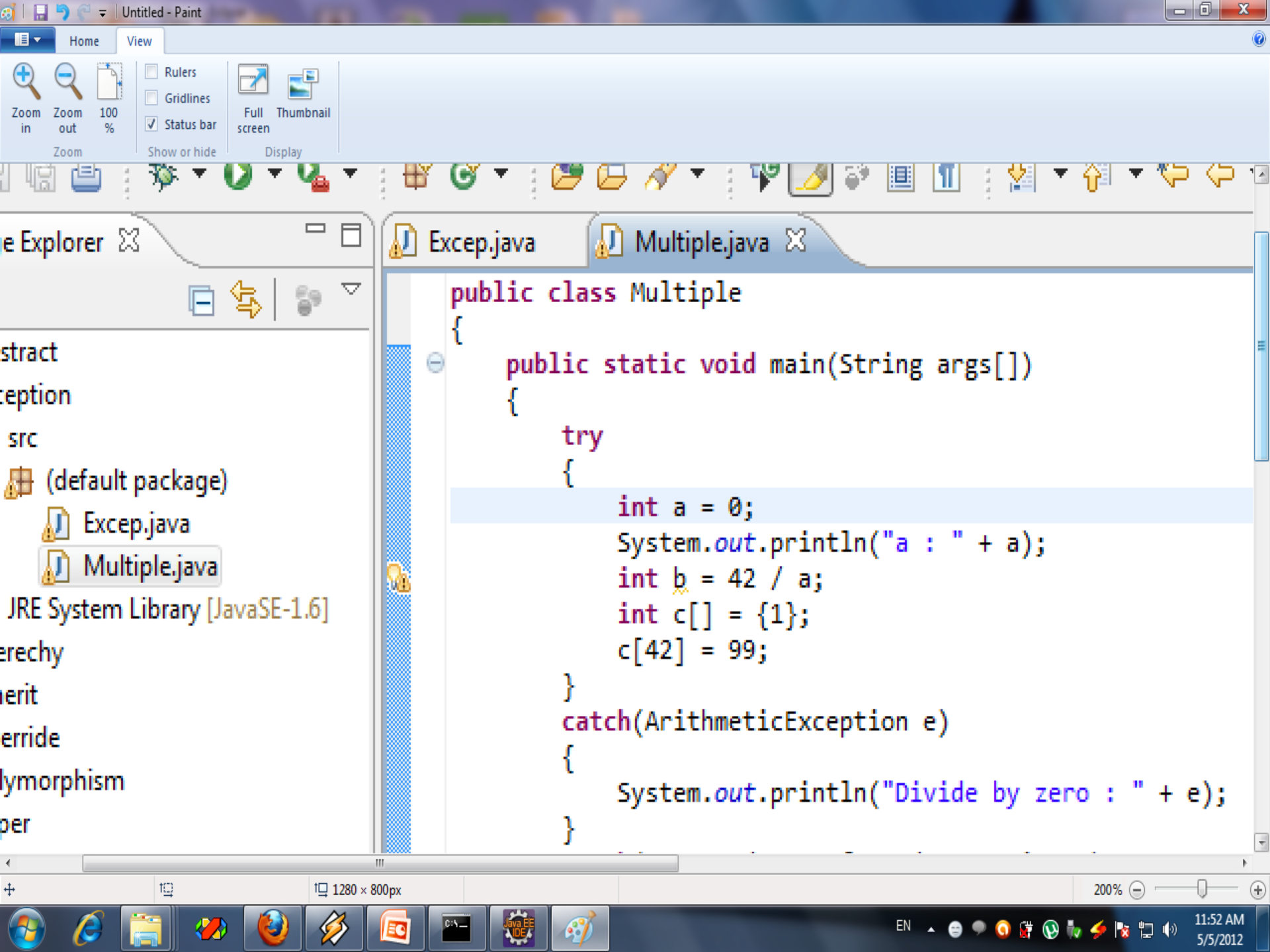
- Now the examples
  1. using try-catch and our own handler
  2. ***Using try-catch and printing default string corresponding to the exception.***
  3. Multiple catch clauses
  4. Nested try statement

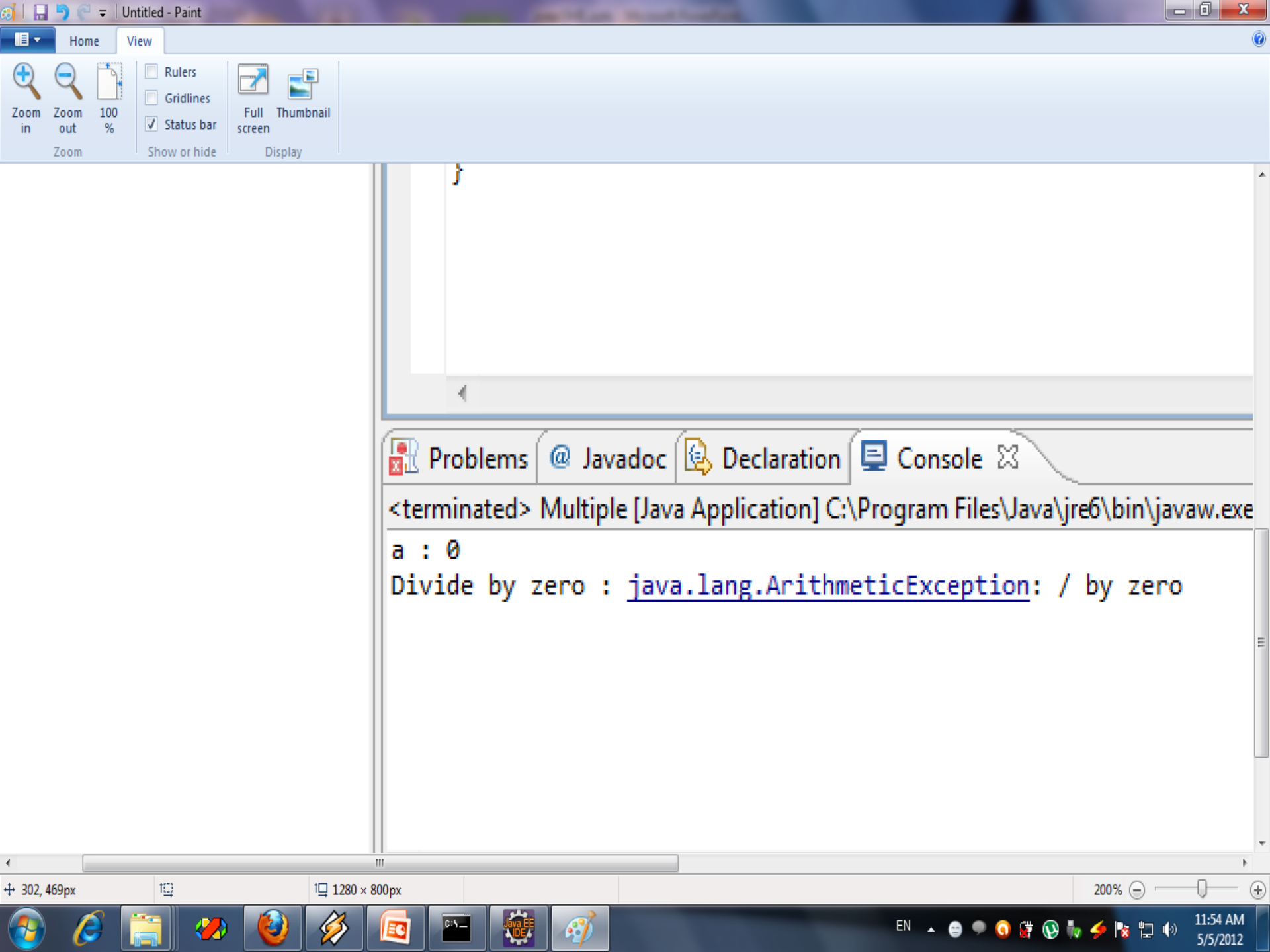

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Excep.java
public class Excep
{
public static void main(String args[])
{
int d, a;
try
{
d = 0;
a = 42 / d;
System.out.println("This will not be printed");
}
catch(ArithmeticException e)
{
System.out.println("Division by zero");
System.out.println("Exception : " + e);
}
}
}
(default package)
Excep.java
System Library [JavaSE-1.6]
564, 229px
1280 × 800px
200%
EN
11:36 PM
5/4/2012

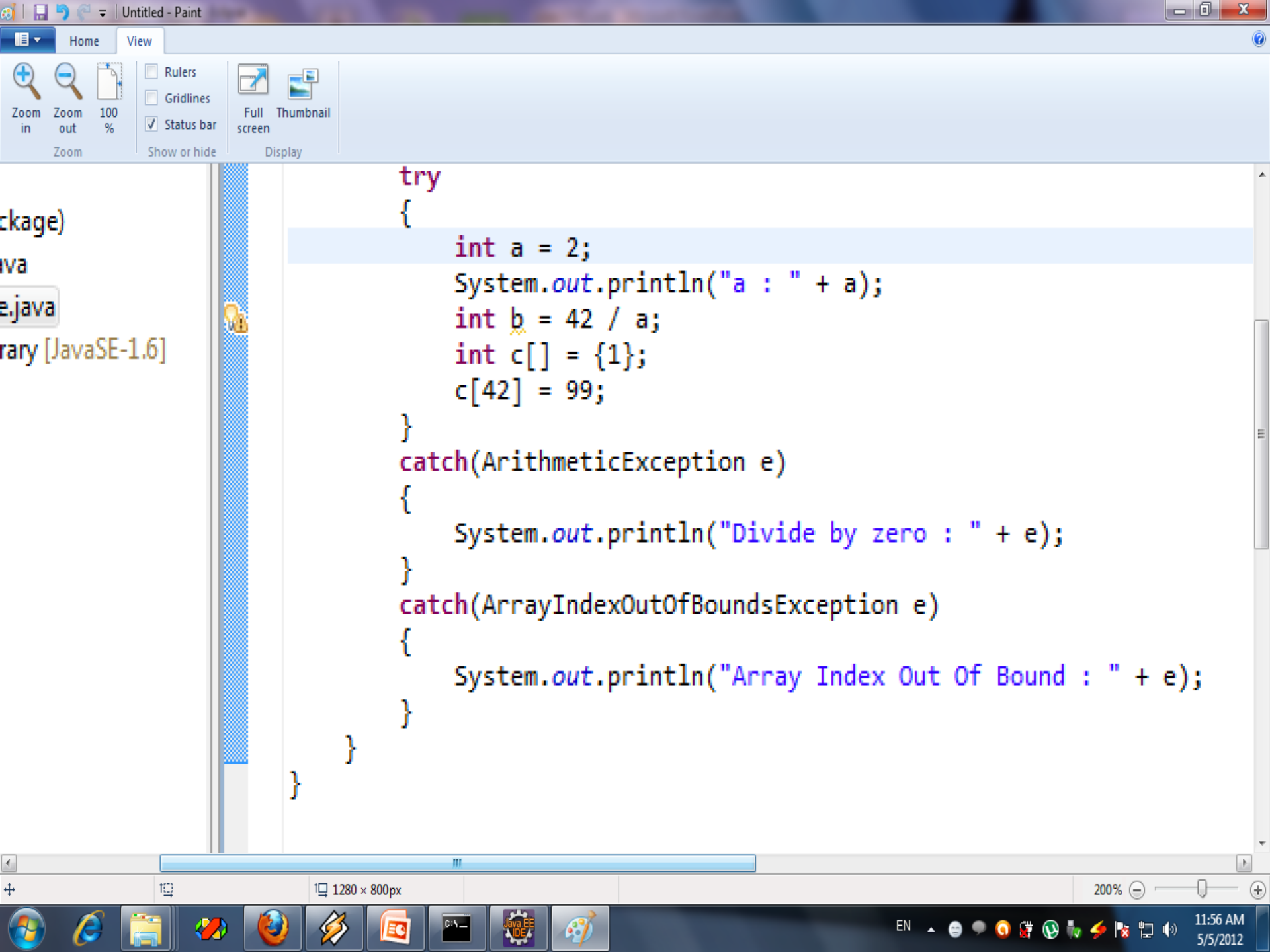

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> Excep [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 4, 2012 11:35:
Division by zero
Exception : java.lang.ArithmeticException: / by zero
1280 × 800px
200%
EN
11:36 PM
5/4/2012

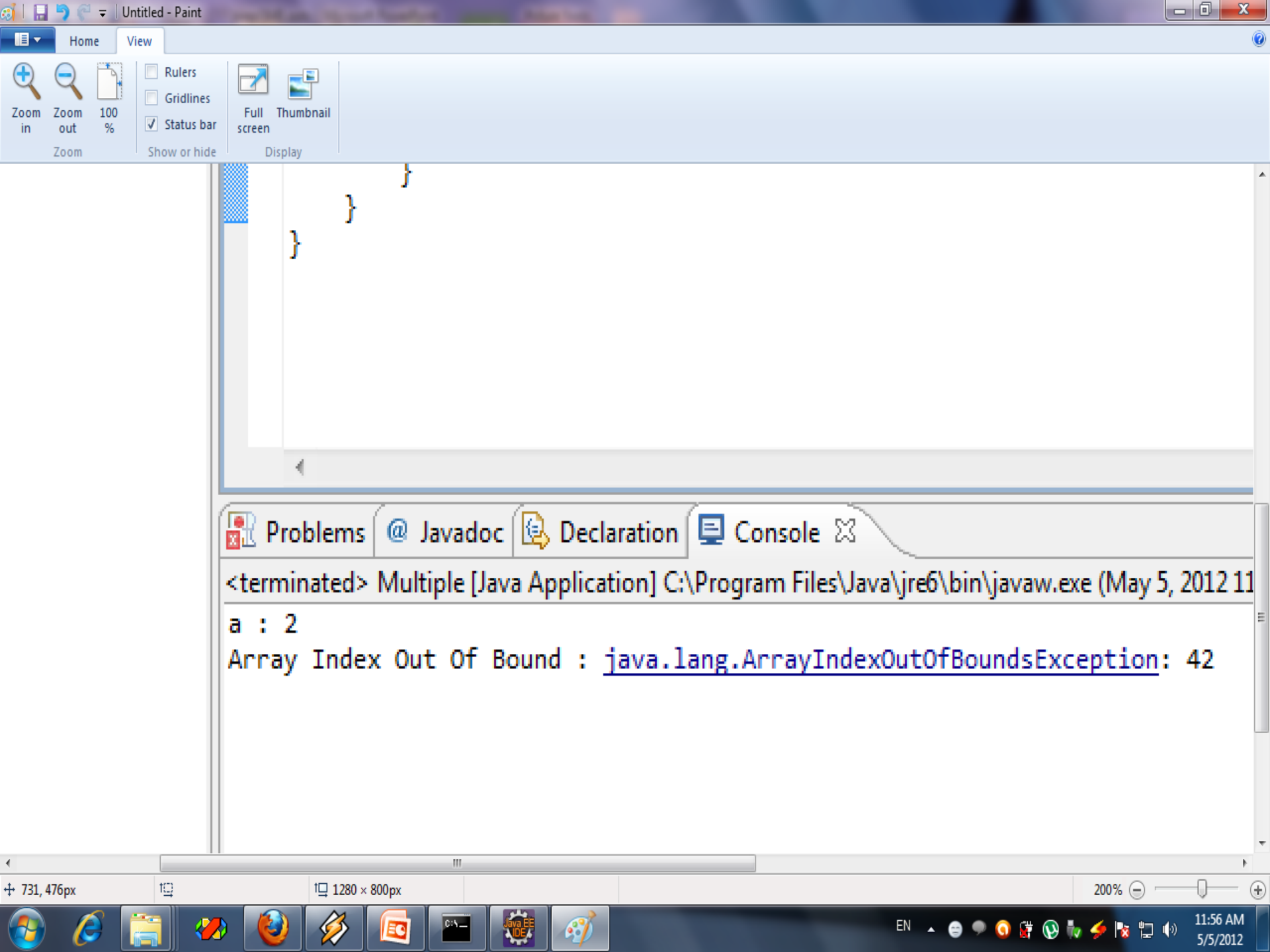
Using Try-Catch()

- Now the examples
  1. using try-catch and our own handler
  2. Using try-catch and printing default string corresponding to the exception.
  3. ***Multiple catch clauses***
  4. Nested try statement


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
e Explorer
stract
ception
src
(default package)
Excep.java
Multiple.java
JRE System Library [JavaSE-1.6]
erechy
herit
erride
lymorphism
per
Excep.java
Multiple.java
public class Multiple
{
public static void main(String args[])
{
try
{
int a = 0;
System.out.println("a : " + a);
int b = 42 / a;
int c[] = {1};
c[42] = 99;
}
catch(ArithmeticException e)
{
System.out.println("Divide by zero : " + e);
}
1280 × 800px
200%
EN
11:52 AM
5/5/2012

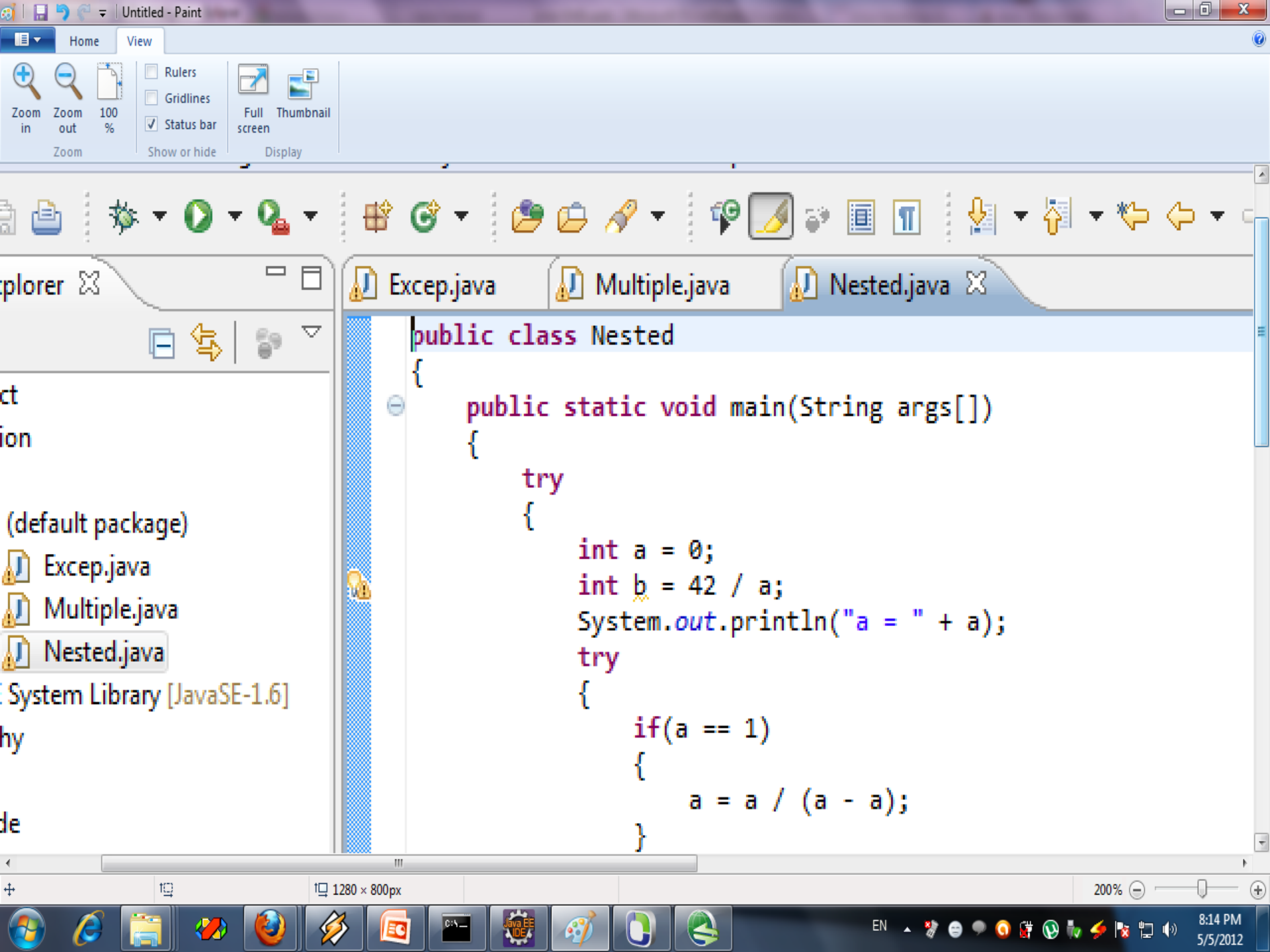

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
}
Problems
Javadoc
Declaration
Console
<terminated> Multiple [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe
a : 0
Divide by zero : java.lang.ArithmeticException: / by zero
302, 469px
1280 × 800px
200%
EN
11:54 AM
5/5/2012

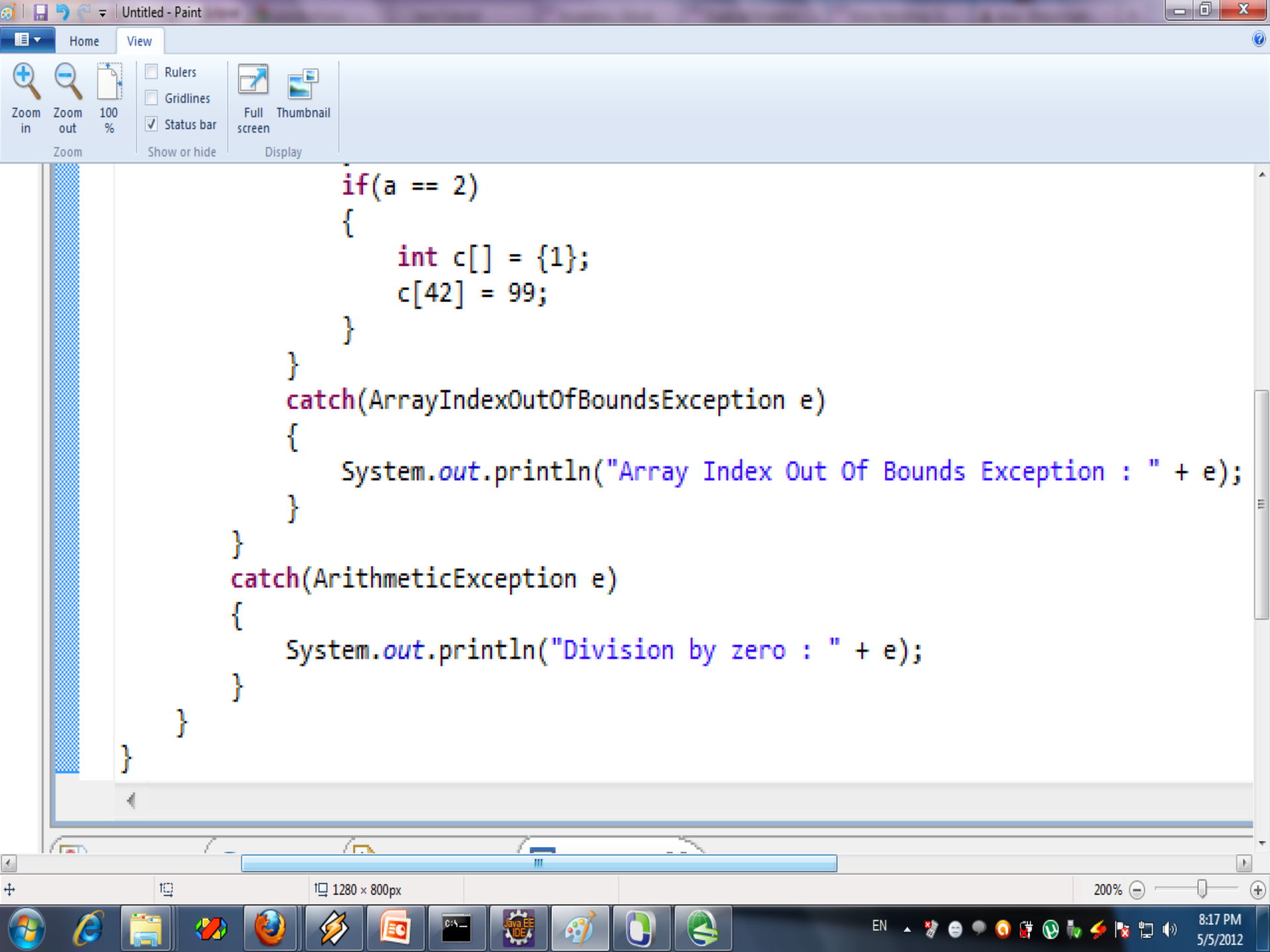

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
ckage)
ava
e.java
rary [JavaSE-1.6]
try
{
int a = 2;
System.out.println("a : " + a);
int b = 42 / a;
int c[] = {1};
c[42] = 99;
}
catch(ArithmeticException e)
{
System.out.println("Divide by zero : " + e);
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("Array Index Out Of Bound : " + e);
}
}
}
1280 × 800px
200%
EN
11:56 AM
5/5/2012

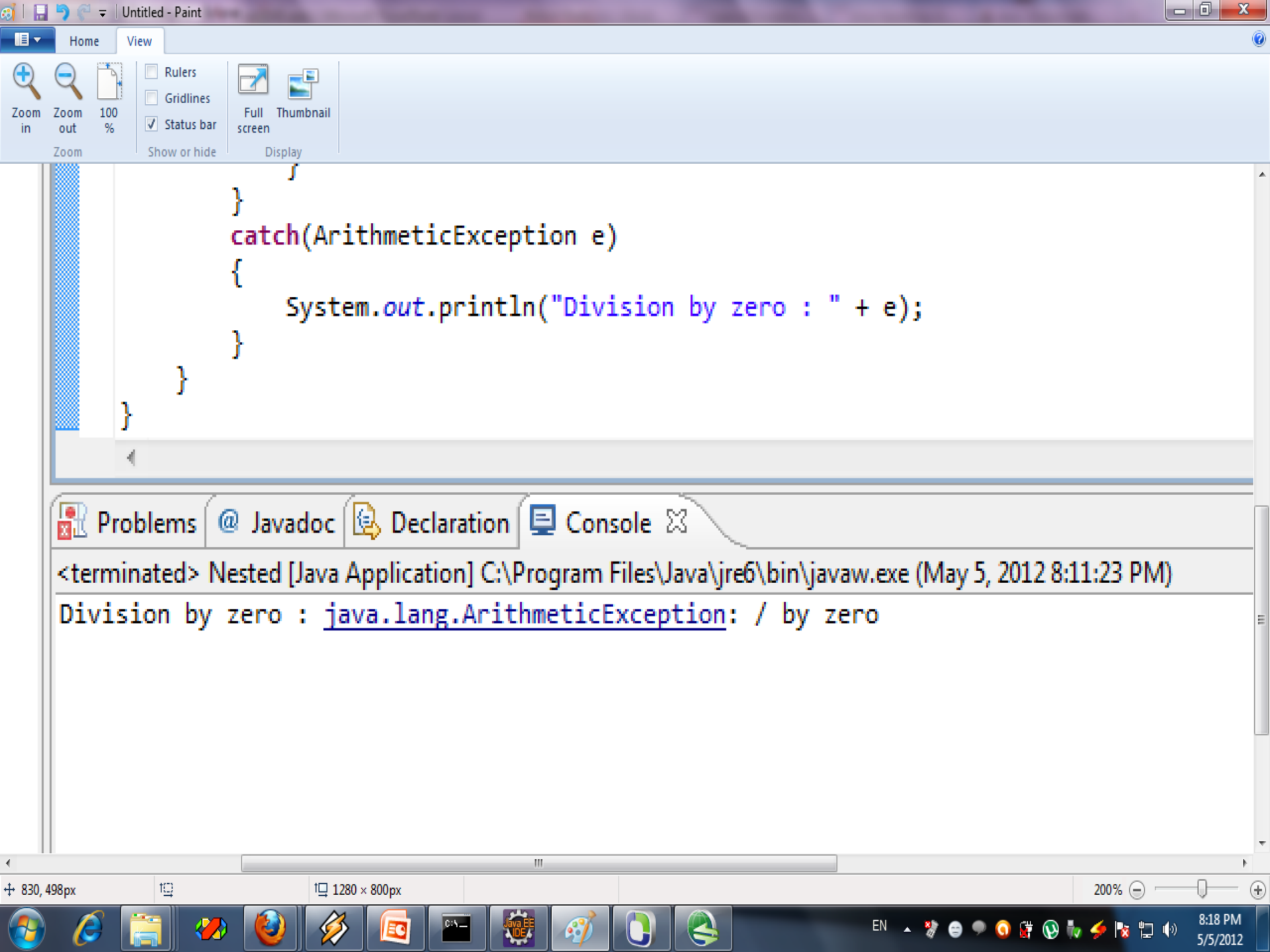

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
}
}
}
Problems
Javadoc
Declaration
Console
<terminated> Multiple [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 5, 2012 11
a : 2
Array Index Out Of Bound : java.lang.ArrayIndexOutOfBoundsException: 42
731, 476px
1280 × 800px
200%
EN
11:56 AM
5/5/2012

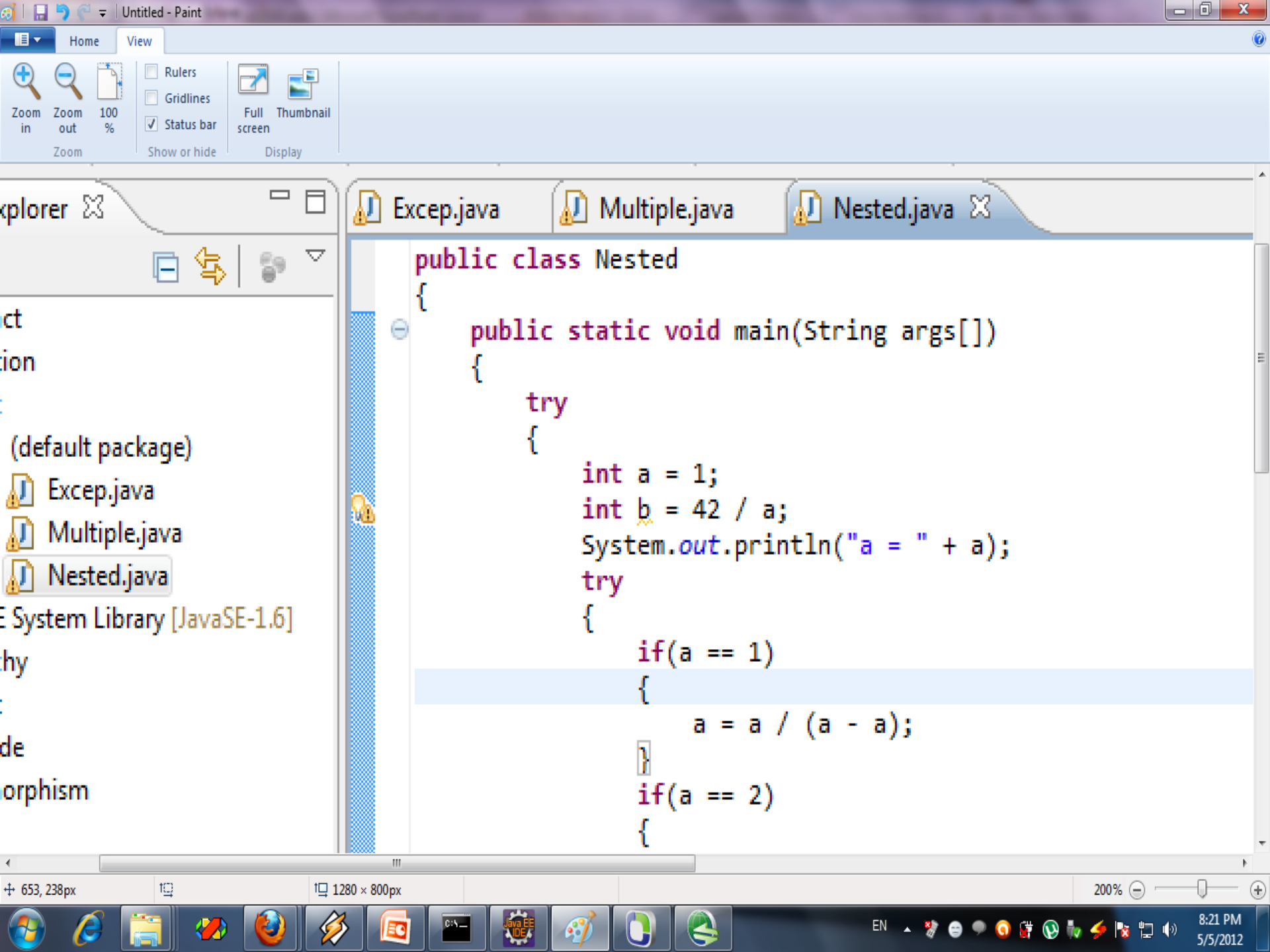
Using Try-Catch()

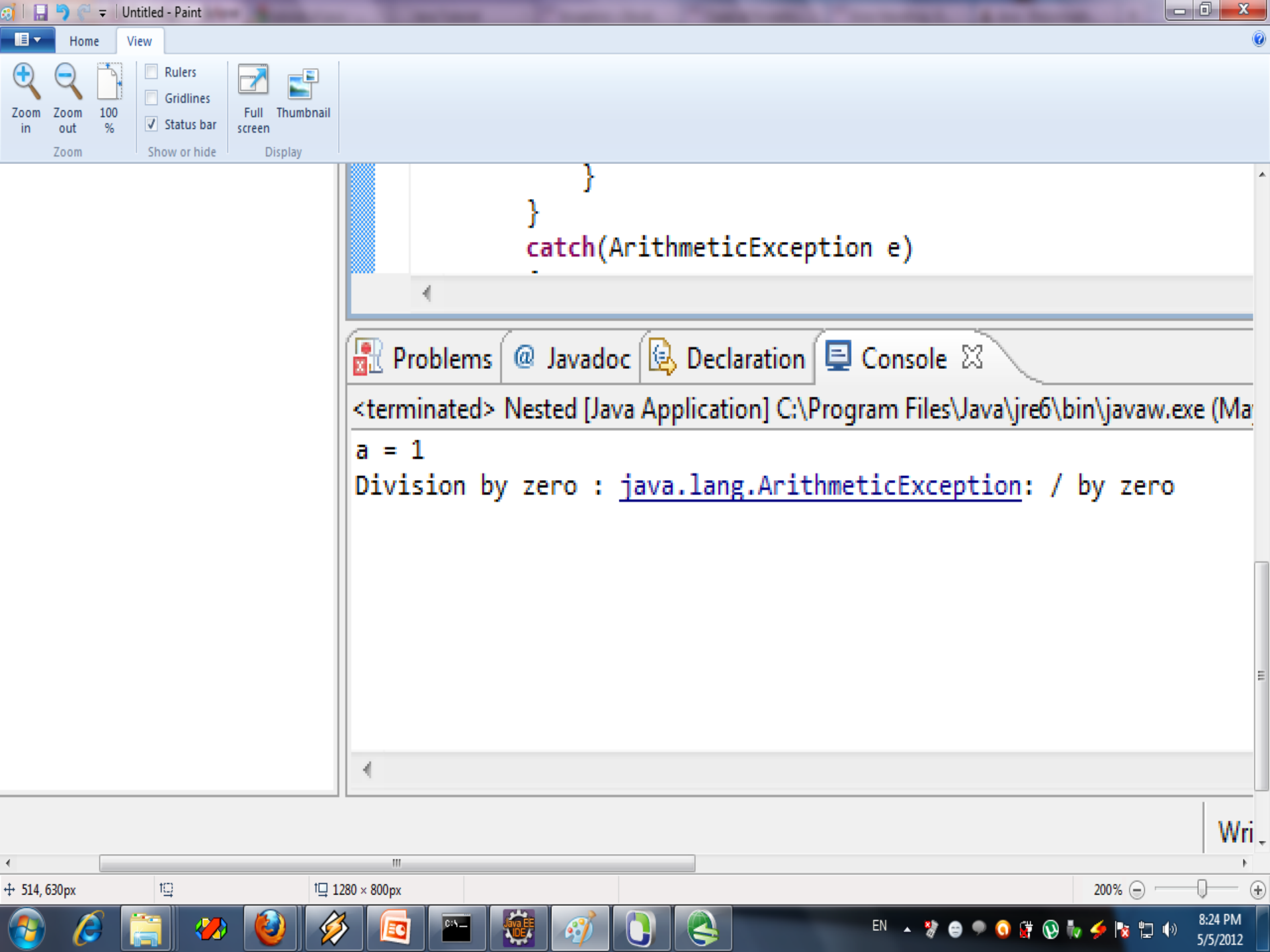
- Now the examples
  1. using try-catch and our own handler
  2. Using try-catch and printing default string corresponding to the exception.
  3. Multiple catch clauses
  4. ***Nested try statement***
  5. Using Throw

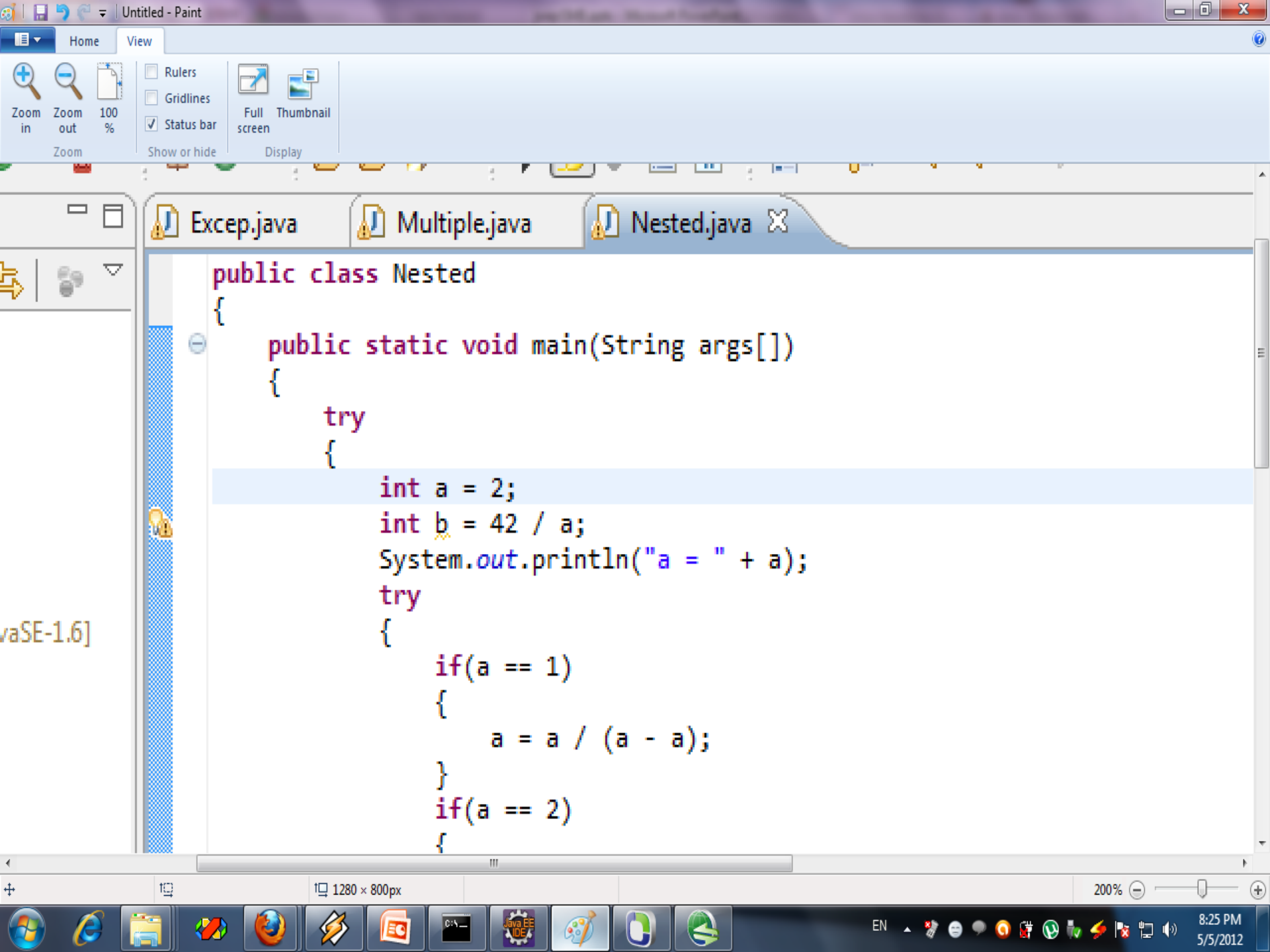

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Excep.java
Multiple.java
Nested.java
public class Nested
{
public static void main(String args[])
{
try
{
int a = 0;
int b = 42 / a;
System.out.println("a = " + a);
try
{
if(a == 1)
{
a = a / (a - a);
}
(default package)
Excep.java
Multiple.java
Nested.java
System Library [JavaSE-1.6]
1280 × 800px
200%
EN
8:14 PM
5/5/2012

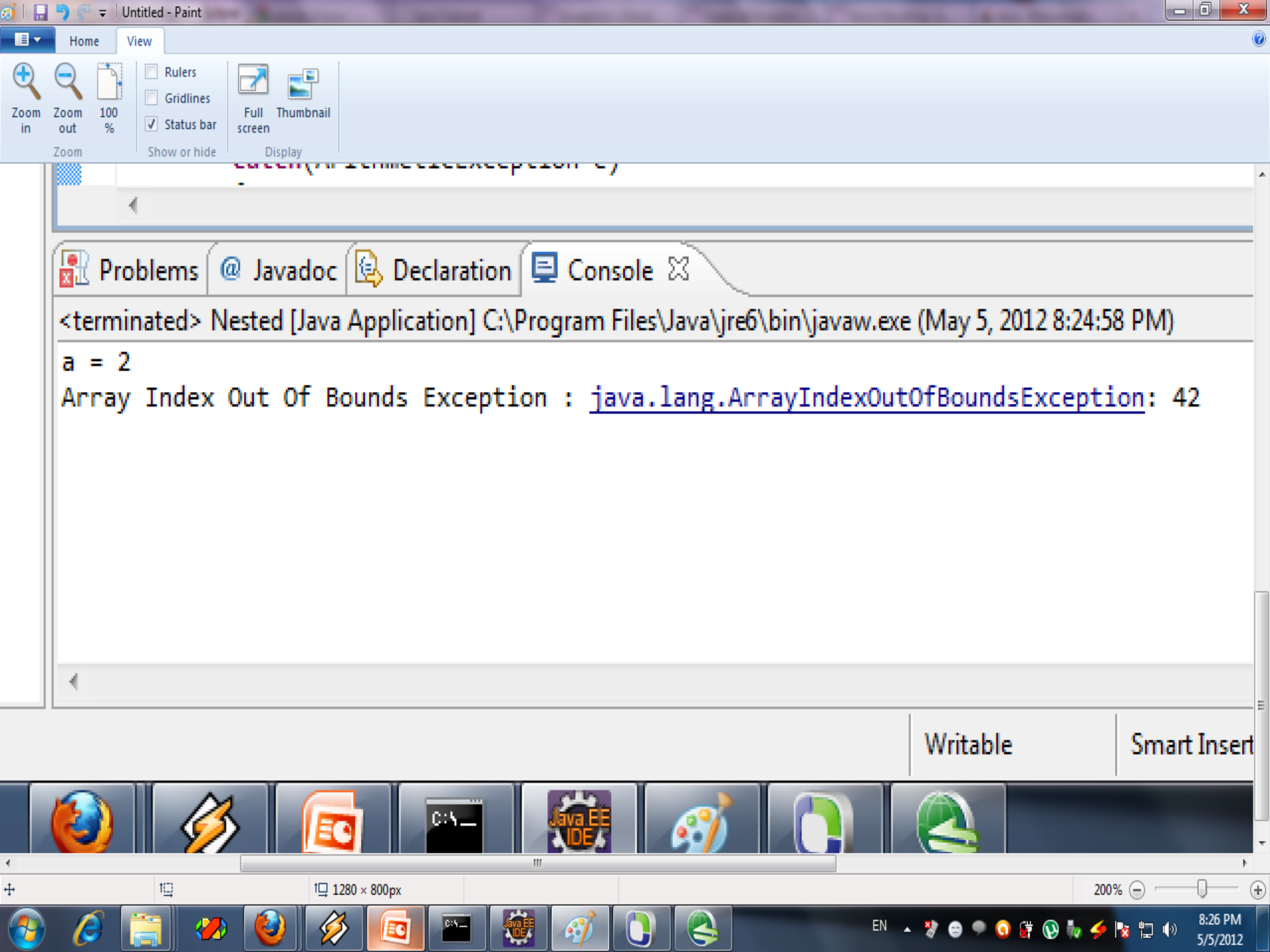

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
if(a == 2)
{
int c[] = {1};
c[42] = 99;
}
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("Array Index Out Of Bounds Exception : " + e);
}
}
catch(ArithmeticException e)
{
System.out.println("Division by zero : " + e);
}
}
}
1280 × 800px
200%
EN
8:17 PM
5/5/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
}
}
catch(ArithmeticException e)
{
System.out.println("Division by zero : " + e);
}
}
}
Problems
Javadoc
Declaration
Console
<terminated> Nested [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 5, 2012 8:11:23 PM)
Division by zero : java.lang.ArithmeticException: / by zero
830, 498px
1280 × 800px
200%
EN
8:18 PM
5/5/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
xplorer
ct
tion
(default package)
Excep.java
Multiple.java
Nested.java
System Library [JavaSE-1.6]
hy
de
orphism
Excep.java
Multiple.java
Nested.java
public class Nested
{
public static void main(String args[])
{
try
{
int a = 1;
int b = 42 / a;
System.out.println("a = " + a);
try
{
if(a == 1)
{
a = a / (a - a);
}
if(a == 2)
{
653, 238px
1280 × 800px
200%
EN
8:21 PM
5/5/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
}
}
catch(ArithmeticException e)
Problems
Javadoc
Declaration
Console
<terminated> Nested [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (Ma
a = 1
Division by zero : java.lang.ArithmeticException: / by zero
514, 630px
1280 × 800px
200%
EN
8:24 PM
5/5/2012


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Excep.java
Multiple.java
Nested.java
public class Nested
{
public static void main(String args[])
{
try
{
int a = 2;
int b = 42 / a;
System.out.println("a = " + a);
try
{
if(a == 1)
{
a = a / (a - a);
}
if(a == 2)
{
vaSE-1.6]
1280 × 800px
200%
EN
8:25 PM
5/5/2012

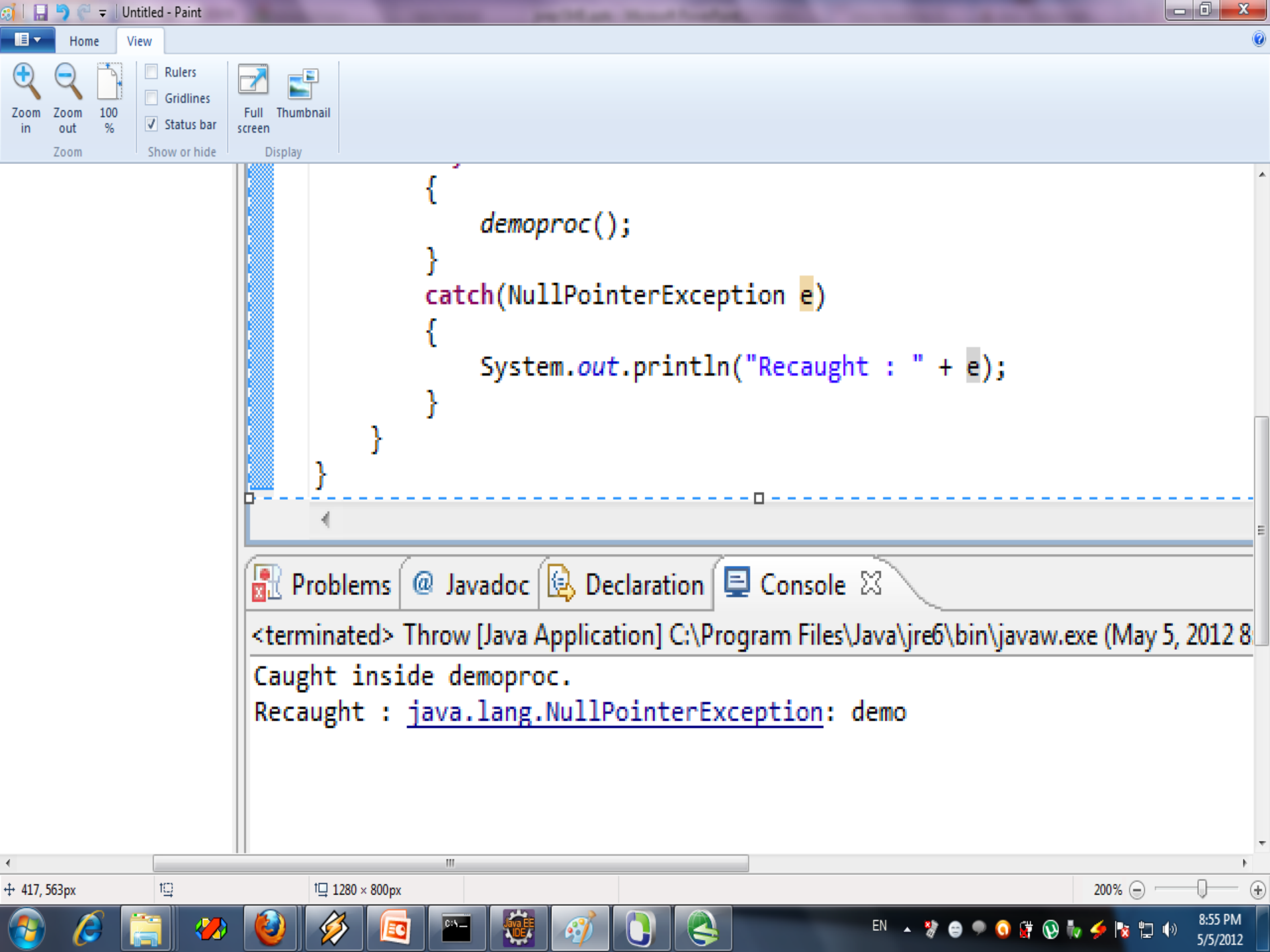

Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
Problems
Javadoc
Declaration
Console
<terminated> Nested [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 5, 2012 8:24:58 PM)
a = 2
Array Index Out Of Bounds Exception : java.lang.ArrayIndexOutOfBoundsException: 42
Writable
Smart Insert
1280 × 800px
200%
EN
8:26 PM
5/5/2012

Using Try-Catch()

- Now the examples
  1. using try-catch and our own handler
  2. Using try-catch and printing default string corresponding to the exception.
  3. Multiple catch clauses
  4. Nested try statement
  5. ***Using Throw***

Excep.java | Multiple.java | Nested.java | Throw.java

```
public class Throw
{
    static void demoproc()
    {
        try
        {
            throw new NullPointerException("demo");
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught inside demoproc.");
            throw e;
        }
    }
    public static void main(String args[])
    {
        try
        {
            demoproc();
        }
        catch(NullPointerException e)
        {
            System.out.println("Recaught : " + e);
        }
    }
}
```


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
{
demoproc();
}
catch(NullPointerException e)
{
System.out.println("Recaught : " + e);
}
}
}
Problems
Javadoc
Declaration
Console
<terminated> Throw [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 5, 2012 8
Caught inside demoproc.
Recaught : java.lang.NullPointerException: demo
417, 563px
1280 × 800px
200%
EN
8:55 PM
5/5/2012

Throws

- If a method is capable of causing an exception that it does not handle, it must specify this behavior so that callers of the method can guard themselves against that exception.
- A ***throws*** clauses lists the types of exceptions that a method might throw.
- This is necessary for all exceptions, except those of type ***error*** or ***RuntimeException***, or any of their subclasses.

Throws

- All other exceptions that a method can throw must be declared in the ***throws*** clauses.
- If they are not, a compile-time error will result.

Excep.java | Multiple.java | Nested.java | Throw.java | Throws.java

```
public class Throws
{
    static void Throws() throws IllegalAccessException
    {
        System.out.println("Inside Throws");
        throw new IllegalAccessException("demo");
    }
    public static void main(String args[])
    {
        try
        {
            Throws();
        }
        catch(IllegalAccessException e)
        {
            System.out.println("caught : " + e);
        }
    }
}
```

Problems | Javadoc | Declaration | Console

<terminated> Throw [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 5, 2012 8:51:09 PM)

```
Caught inside demoproc.
Recaught : java.lang.NullPointerException: demo
```

Finally

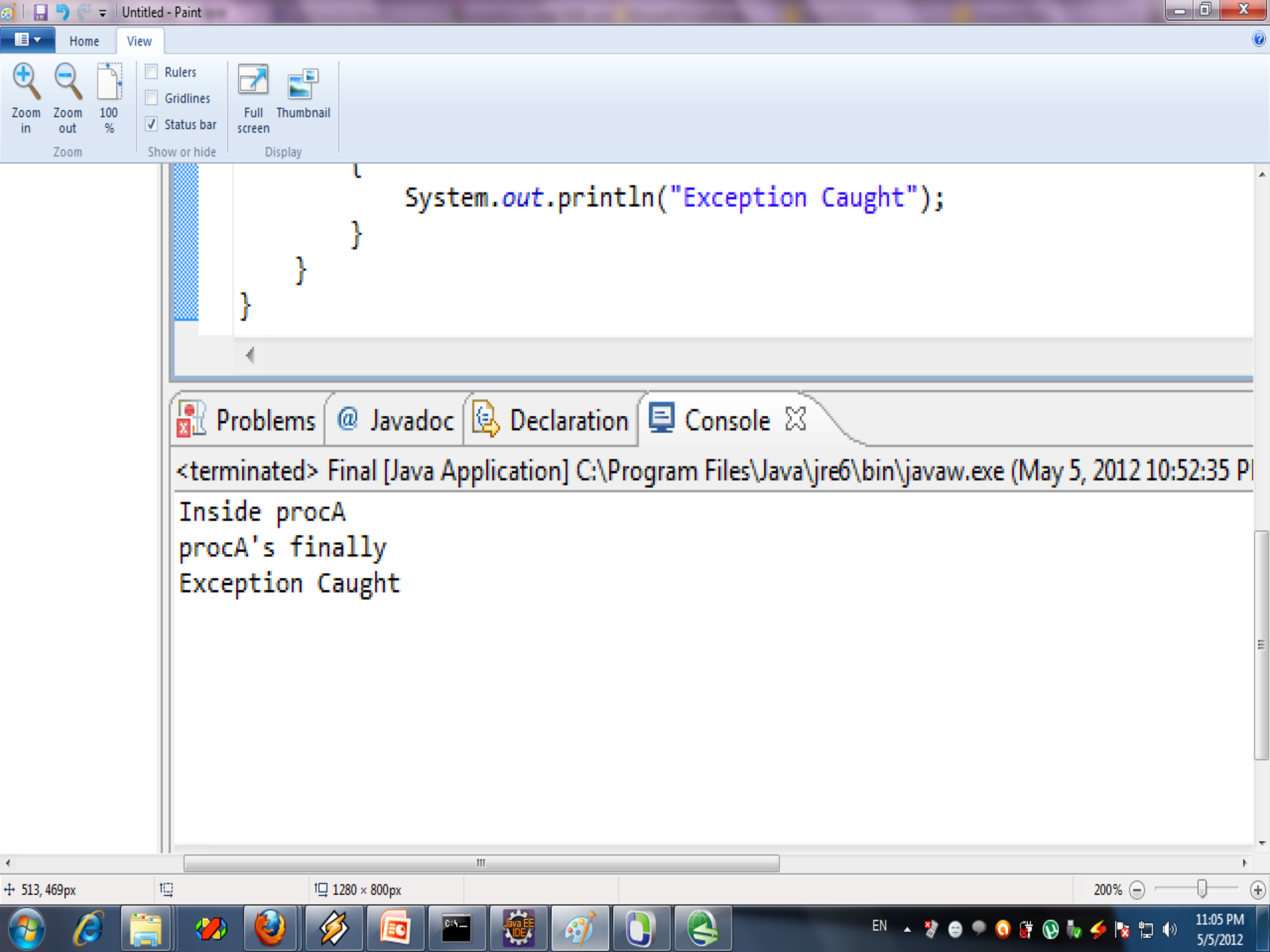
- ***Finally*** creates a block of code that will be executed after a ***try/catch*** block has completed and before the code following the ***try/catch*** block.
- The ***finally*** block will execute whether or not an exception is thrown.
- If an exception is thrown, the ***finally*** block will execute even if no ***catch*** statement matches the exception.

Finally

- The ***finally*** clause is optional.
- However, each ***try*** statement requires at least one ***catch*** or a ***finally*** clauses.

Final.java

```
public class Final
{
    static void procA()
    {
        try
        {
            System.out.println("Inside procA");
            throw new RuntimeException("demo");
        }
        finally
        {
            System.out.println("procA's finally");
        }
    }
    public static void main(String args[])
    {
        try
        {
            procA();
        }
        catch(Exception e)
        {
            System.out.println("Exception Caught");
        }
    }
}
```


Untitled - Paint
Home
View
Zoom in
Zoom out
100 %
Zoom
Rulers
Gridlines
Status bar
Show or hide
Full screen
Thumbnail
Display
System.out.println("Exception Caught");
Problems
Javadoc
Declaration
Console
<terminated> Final [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (May 5, 2012 10:52:35 PM
Inside procA
procA's finally
Exception Caught
513, 469px
1280 × 800px
200%
EN
11:05 PM
5/5/2012

END