

```
Source History
1 public class VariableScopeExample {
2
3     private static int x = 1; // Class variable accessible in the whole class
4
5     public static void main(String[] args) {
6         int y = 5;
7         System.out.println("Class variable x: " + x);
8         System.out.println("Local variable y: " + y);
9         someMethod();
10    }
11
12    public static void someMethod() {
13        System.out.println("Class variable x from someMethod: " + x);
14        // Error: y cannot be accessed here
15        // System.out.println("Local variable y from someMethod: " + y);
16    }
17 }
```

Output - Programming (run)

```
run:
Class variable x: 1
Local variable y: 5
Class variable x from someMethod: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

Start Page x

Apache NetBeans

My NetBeans

Recent Projects

<no recent projects>

Proj... x Services Files

Application6
Application8
Source Packages
javaapplication8
JavaApplication8.java
libraries

Car.java x VariableScopeExample.java x


```
Source History
3 // Fields (attributes)
4 private String color;
5 private String model;
6
7 // Constructor
8 public Car(String color, String model) {
9     this.color = color;
10    this.model = model;
11 }
12
13 // Method to display information about the car
14 public void displayInfo() {
15     System.out.println("Car model: " + model + ", Color: " + color);
16 }
17
18 // Main method to create and use objects of Car class
19 public static void main(String[] args) {
20     Car myCar = new Car("Red", "Toyota Corolla");
21     myCar.displayInfo();
22 }
23 }
```

Output - Programming (run)

```
run:
Car model: Toyota Corolla, Color: Red
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output

Start Page ×

 Learn 8

My NetBeans

Recent Projects

<no recent project>

Projects × Services Files

JDK 21 (Default)

Test Libraries

JavaApplication2

JavaApplication6

JavaApplication8

Source Packages

javasapplication8

Output - Programming (run)

run:
Int value: 9
Converted to double: 9.0
Double value: 9.78
Converted to int: 9
BUILD SUCCESSFUL (total time: 0 seconds)

Car.java × VariableScopeExample.java × TypeCasting.java ×

Source History

```
1 public class TypeCasting {  
2  
3     public static void main(String[] args) {  
4  
5         // Implicit casting (automatic type conversion)  
6         int myInt = 9;  
7         double myDouble = myInt;  
8  
9         System.out.println("Int value: " + myInt);  
10        System.out.println("Converted to double: " + myDouble);  
11  
12        // Explicit casting (manual type conversion)  
13        double anotherDouble = 9.78;  
14        int anotherInt = (int) anotherDouble;  
15  
16        System.out.println("Double value: " + anotherDouble);  
17        System.out.println("Converted to int: " + anotherInt);  
18    }  
19 }
```

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Programming - Apache NetBeans IDE 21 Search (Ctrl+I)

<default config> 377.4/529.0MB Resume (pending events: 0)

Start Page x

Apache NetBeans IDE Learn 8

My NetBeans

Recent Projects

<no recent project>

Projects x Services Files

- JavaApplication8
 - Source Packages
 - javaapplication8
 - JavaApplication8.java
 - Libraries
 - JavaApplication9
 - Programing

Output - Programming (run)

```
run:
An exception occurred: Index 5 out of bounds for length 3
The 'try catch' is finished
BUILD SUCCESSFUL (total time: 0 seconds)
```

Source History

```
1 public class ExceptionHandlingExample {
2
3     public static void main(String[] args) {
4
5         try {
6             int[] numbers = {1, 2, 3};
7             System.out.println(numbers[5]); // This will throw an ArrayIndexOutOfBoundsException
8         } catch (ArrayIndexOutOfBoundsException e) {
9             System.out.println("An exception occurred: " + e.getMessage());
10        } finally {
11            System.out.println("The 'try catch' is finished");
12        }
13    }
14 }
15 }
```

15:2 INS Windows (CRLF) 11:32 02-05-2024

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Programming - Apache NetBeans IDE 21 Search (Ctrl+I)

<default config> 339.4/580.0MB Resume (pending events: 5)

Start Page × NewClass.java × NewClass.java × VariableScopeExample.java × Car.java × TypeCasting.java × ExceptionHandlingExample.java × Main.java ×

Source History

```
1 class Vehicle { // Base class (parent)
2     protected String brand = "Ford"; // Vehicle attribute
3
4     public void honk() { // Vehicle method
5         System.out.println("Tut tut!");
6     }
7 }
8
9 class Car extends Vehicle { // Derived class (child)
10     private String modelName = "Mustang"; // Car attribute
11
12     public void displayModel() {
13         System.out.println("Brand: " + brand + ", Model: " + modelName);
14     }
15 }
16
17 public class Main {
18     public static void main(String[] args) {
19         Car myCar = new Car();
20         myCar.honk();
21         myCar.displayModel();
22     }
23 }
```

Recent Projects: <no recent projects>

Output - Programming (run)

```
run:
Tut tut!
Brand: Ford, Model: Mustang
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output

16:1 INS 32°C Sunny 11:36 02-05-2024

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Programming - Apache NetBeans IDE 21

Search (Ctrl+I)

329.176010MB

Resume (pending events: 4)

Start Page x

VariableScopeExample.java x Car.java x TypeCasting.java x ExceptionHandlingExample.java x Main.java x ain.java x

Source History

```
11 }
12 }
13
14 class Cat extends Animal {
15     @Override
16     public void sound() {
17         System.out.println("Meow");
18     }
19 }
20
21 public class ain {
22     public static void main(String[] args) {
23         Animal myAnimal = new Animal();
24         Animal myDog = new Dog(); // Corrected syntax: '=' instead of '-'
25         Animal myCat = new Cat();
26
27         myAnimal.sound(); // Outputs Some sound
28         myDog.sound();    // Outputs Woof
29         myCat.sound();    // Outputs Meow
30     }
31 }
```

Output - Programming (run)

run:
Some sound
Woof
Meow
BUILD SUCCESSFUL (total time: 0 seconds)

Output

31:2 INS

Type here to search

32°C Sunny 11:40 02-05-2024

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Programming - Apache NetBeans IDE 21 Search (Ctrl+I)

Start Page x VariableScopeExample.java x Car.java x TypeCasting.java x ExceptionHandlingExample.java x Main.java x ain.java x Demo.java x

Source History

```
1 class AccessSpecifierDemo {
2
3     public int publicVar = 100; // Accessible from any other class
4     private int privateVar = 200; // Accessible only within the class
5     protected int protectedVar = 300; // Accessible within the class and by derived classes
6
7     public void display() {
8         System.out.println("Public: " + publicVar);
9         System.out.println("Private: " + privateVar);
10        System.out.println("Protected: " + protectedVar);
11    }
12 }
13
14 public class Demo {
15     public static void main(String[] args) {
16         AccessSpecifierDemo demo = new AccessSpecifierDemo();
17         demo.display();
18         System.out.println("PublicVar: " + demo.publicVar);
19         // System.out.println("PrivateVar: " + demo.privateVar); // Error: Cannot access private variable
20         // System.out.println("ProtectedVar: " + demo.protectedVar); // Error: Cannot access protected variable outside the class without
21     }
22 }
```

Output - Programming (run)

```
Run:
Public: 100
Private: 200
Protected: 300
PublicVar: 100
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output

14:20 INS 32°C Sunny 11:42 02-05-2024

Start Page x

Apache NetBeans My NetBeans Recent Projects <no recent projects>

Proj... x Services Files

Source Packages <default package> Car.java Car1.java Car1Impl.java Demo.java

Source History

```
1 class Car {
2
3     private String model;
4     private int year;
5
6     // Constructor
7     public Car(String model, int year) {
8         this.model = model;
9         this.year = year;
10    }
11
12    public void displayInfo() {
13        System.out.println("Model: " + model + " Year: " + year);
14    }
15 }
16
17 public class Car1 {
18     public static void main(String[] args) {
19         Car car1 = new Car("Toyota Corolla", 2020);
20         car1.displayInfo();
21     }
22 }
```

Output - Programming (run)

```
run:
Model: Toyota Corolla Year: 2020
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
1 interface Animal {
2     void sound(); // Interface method (does not have a body)
3 }
4
5 class Dog implements Animal {
6     public void sound() {
7         // The body of sound() is provided here
8         System.out.println("Woof");
9     }
10 }
11
12 class Cat implements Animal {
13     public void sound() {
14         System.out.println("Meow");
15     }
16 }
17
18 public class Klion{
19     public static void main(String[] args) {
20         Dog myDog = new Dog();
21         Cat myCat = new Cat(); // Corrected syntax: added parentheses
22         myDog.sound();
23         myCat.sound();
24     }
25 }
```

Output - Programming (run)

```
Run:
Woof
Meow
BUILD SUCCESSFUL (total time: 0 seconds)
```

Start Page x

Apache NetBeans

My NetBeans

Recent Projects

<no recent projects>

Proj... x Services Files

- JavaApplication1
- Libraries
- JavaApplication9
- Programming
- Source Packages
 - <default package>
- Car.java

Source History

```
1 class ThreadDemo extends Thread {
2     public void run() {
3         try {
4             // Moving thread to Timed Waiting state
5             Thread.sleep(150);
6         } catch (InterruptedException e) {
7             e.printStackTrace();
8         }
9         System.out.println("State after completion: " + Thread.currentThread().getState());
10    }
11
12    public static void main(String[] args) throws InterruptedException {
13        ThreadDemo t1 = new ThreadDemo();
14        System.out.println("State when created: " + t1.getState());
15        t1.start();
16        System.out.println("State when started: " + t1.getState());
17        // waiting for thread to die
18        t1.join();
19        System.out.println("State after thread ended its task: " + t1.getState());
20    }
21 }
```

Output - Programming (run)

```
run:
State when created: NEW
State when started: RUNNABLE
State after completion: RUNNABLE
State after thread ended its task: TERMINATED
BUILD SUCCESSFUL (total time: 0 seconds)
```