Main.java

*Static Nested Class*

**package** main;

**import** outer.\*;

**import** **static** outer.Outer.\*;

**class** A {

**private** **int** a = 1;

**private** **static** **int** *x* = 2;

**static** **class** B {

**private** **int** b = 3;

**private** **static** **int** *y* = 4;

**public** **int** getA() {

**return** **new** A().a; //Has access to private member of A

}

**public** **int** getX() {

**return** *x*; //Has access to private member of A. Also ok as return A.x;

}

}

**public** **int** getB() {

// return B.b; //Compilation fails. Cannot make a static reference to the non-static field B.b

**return** **new** B().b; //Has access to private member of B

}

**public** **int** getY() {

**return** B.*y*; //Has access to private member of B

}

}

**public** **class** Main {

**public** **static** **void** main(String[] args) {

A a = **new** A();

System.***out***.println(a.getB()); //Prints: 3

System.***out***.println(a.getY()); //Prints: 4

//B b = new B(); //Compilation fails. B cannot be resolved to a type

A.B b = **new** A.B();

System.***out***.println(b.getA()); //Prints: 1

System.***out***.println(b.getX()); //Prints: 2

//Testing importing Static Inner Class from Different Package

Outer.Inner oi = **new** Outer.Inner();

//required import:

//import outer.\*; or

//import outer.Outer;

Inner i = **new** Inner();

//required import:

//import outer.Outer.\*; or

//import outer.Outer.Inner; or

//import static outer.Outer.\*; or

//import static outer.Outer.Inner;

}

}

Outer.java

**package** outer;

**public** **class** Outer {

**public** **static** **class** Inner {

}

}