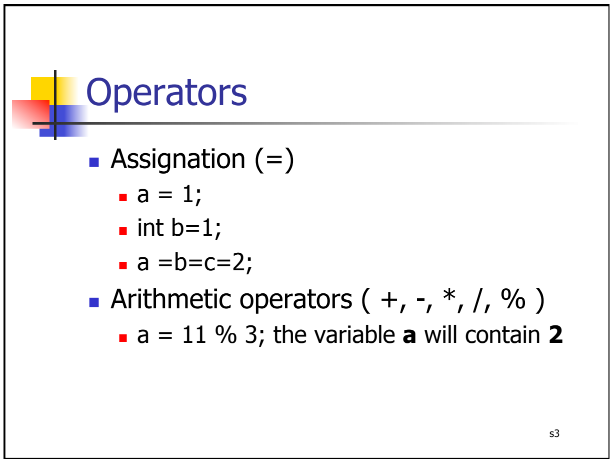
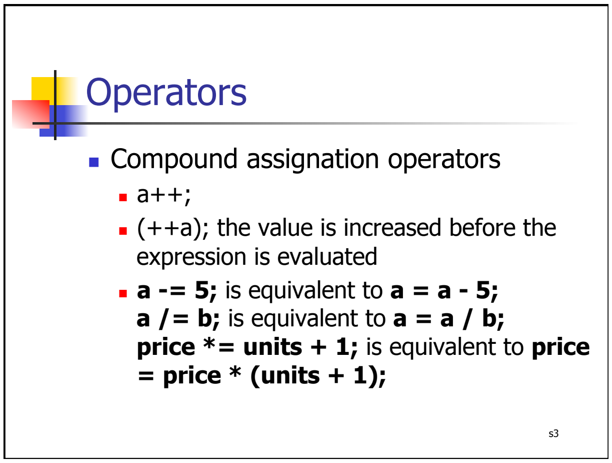


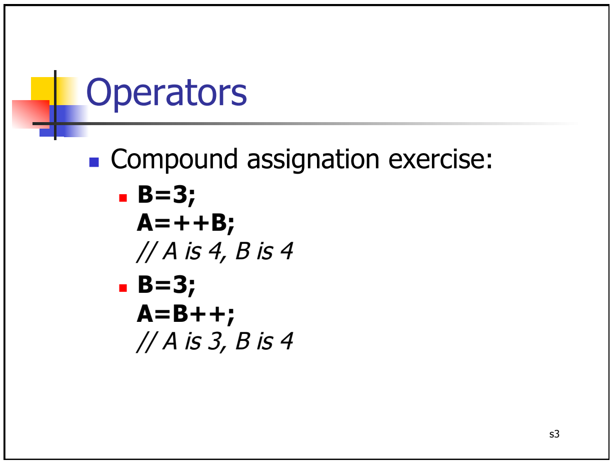
Operators

- Assignation (=)
  - a = 1;
  - int b=1;
  - a =b=c=2;
- Arithmetic operators (+, -, *, /, %)
  - a = 11 % 3; the variable **a** will contain **2**



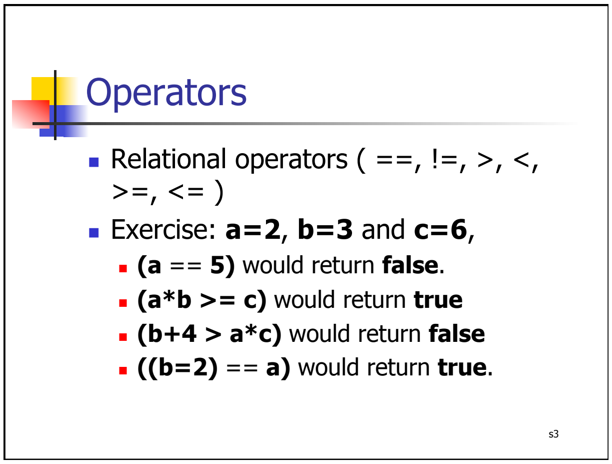
Operators

- Compound assignation operators
  - a++;
  - (++a); the value is increased before the expression is evaluated
  - **a -= 5;** is equivalent to **a = a - 5;**
    a /= b; is equivalent to **a = a / b;**
    price *= units + 1; is equivalent to **price = price * (units + 1);**



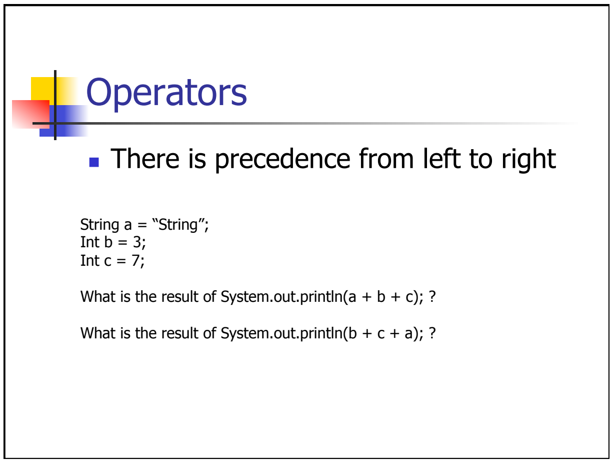
Operators

- Compound assignation exercise:
  - **B=3;**
    A=++B;
    // A is 4, B is 4
  - **B=3;**
    A=B++;
    // A is 3, B is 4



Operators

- Relational operators (==, !=, >, <, >=, <=)
- Exercise: **a=2**, **b=3** and **c=6**,
  - **(a** == **5)** would return **false**.
  - **(a*b >= c)** would return **true**
  - **(b+4 > a*c)** would return **false**
  - **((b=2)** == **a)** would return **true**.



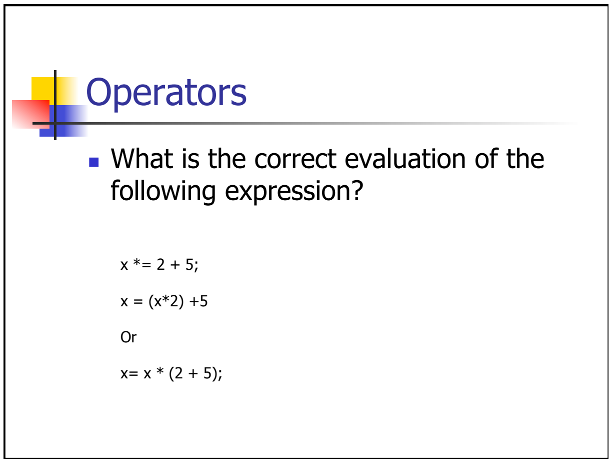
Operators

- There is precedence from left to right

```
String a = "String";
Int b = 3;
Int c = 7;
```

What is the result of System.out.println(a + b + c); ?

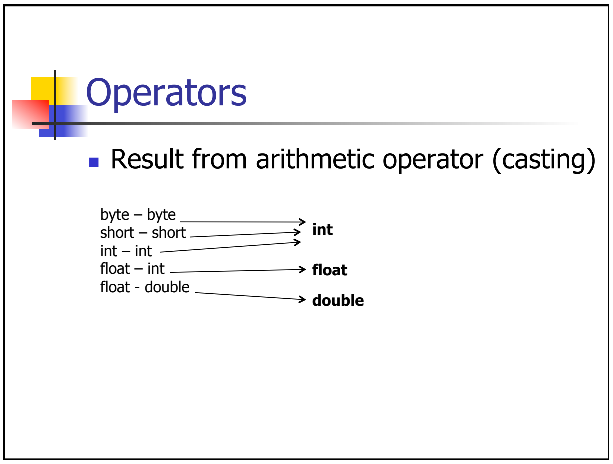
What is the result of System.out.println(b + c + a); ?



Operators

- What is the correct evaluation of the following expression?

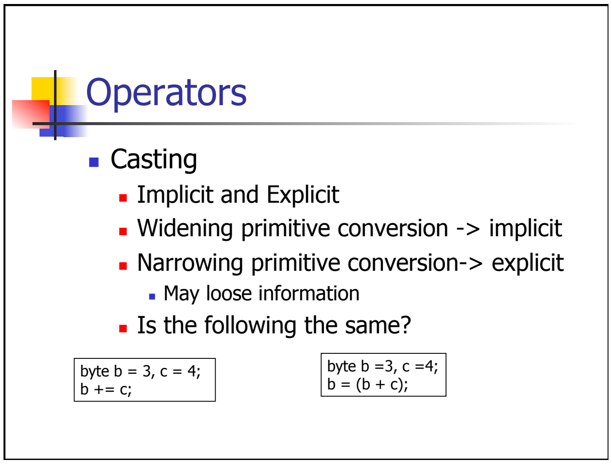
```
x *= 2 + 5;

x = (x*2) +5

Or

x= x * (2 + 5);
```



Operators

- Result from arithmetic operator (casting)

byte – byte → **int**
short – short → **int**
int – int → **int**
float – int → **float**
float - double → **double**

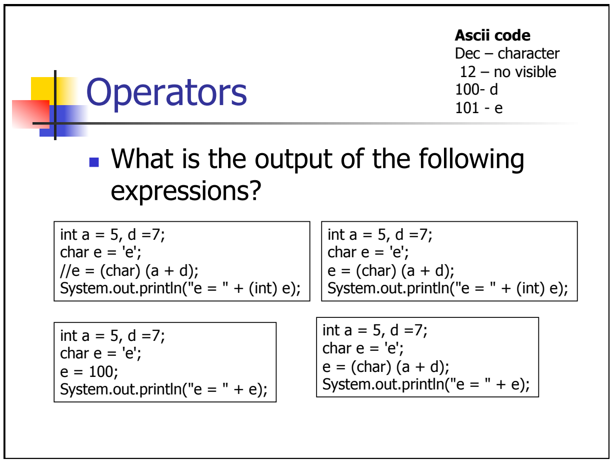


Operators

- Casting
  - Implicit and Explicit
  - Widening primitive conversion -> implicit
  - Narrowing primitive conversion-> explicit
    - May loose information
  - Is the following the same?

```
byte b = 3, c = 4;
b += c;
```

```
byte b =3, c =4;
b = (b + c);
```



Ascii code
Dec – character
12 – no visible
100- d
101 - e

Operators

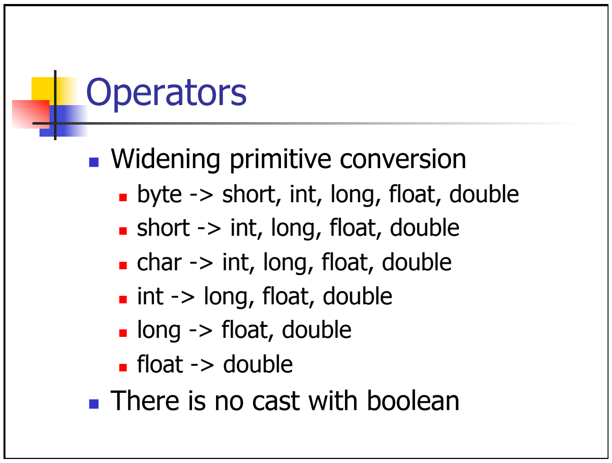
- What is the output of the following expressions?

```
int a = 5, d =7;
char e = 'e';
//e = (char) (a + d);
System.out.println("e = " + (int) e);
```

```
int a = 5, d =7;
char e = 'e';
e = (char) (a + d);
System.out.println("e = " + (int) e);
```

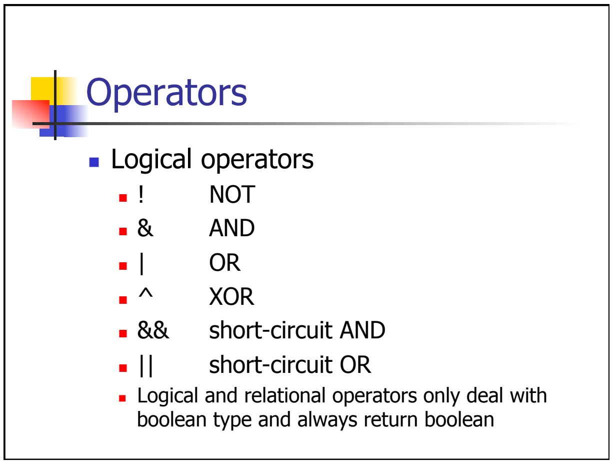
```
int a = 5, d =7;
char e = 'e';
e = 100;
System.out.println("e = " + e);
```

```
int a = 5, d =7;
char e = 'e';
e = (char) (a + d);
System.out.println("e = " + e);
```



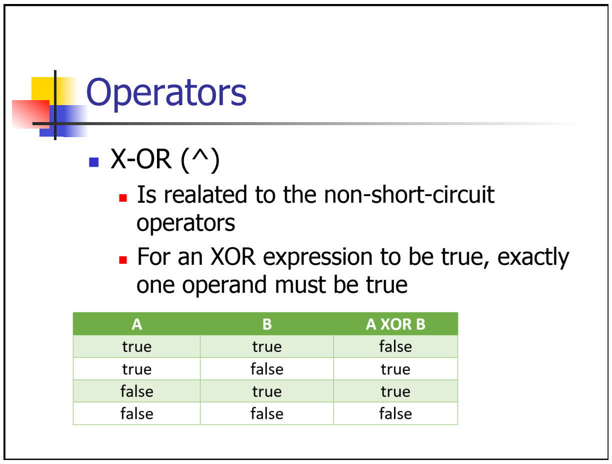
Operators

- Widening primitive conversion
  - byte -> short, int, long, float, double
  - short -> int, long, float, double
  - char -> int, long, float, double
  - int -> long, float, double
  - long -> float, double
  - float -> double
- There is no cast with boolean



Operators

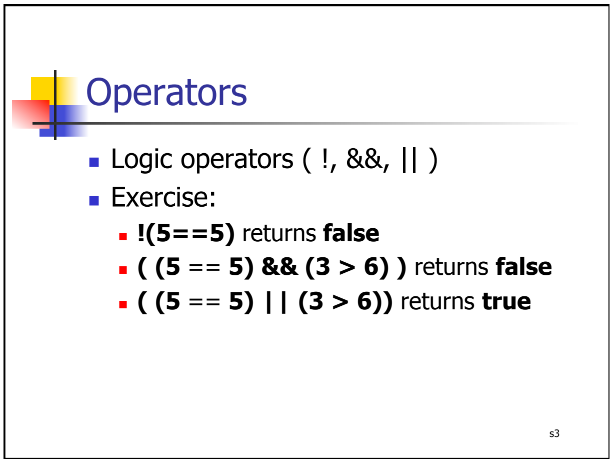
- Logical operators
  - ! NOT
  - & AND
  - | OR
  - ^ XOR
  - && short-circuit AND
  - || short-circuit OR
  - Logical and relational operators only deal with boolean type and always return boolean



Operators

- X-OR (^)
  - Is realated to the non-short-circuit operators
  - For an XOR expression to be true, exactly one operand must be true

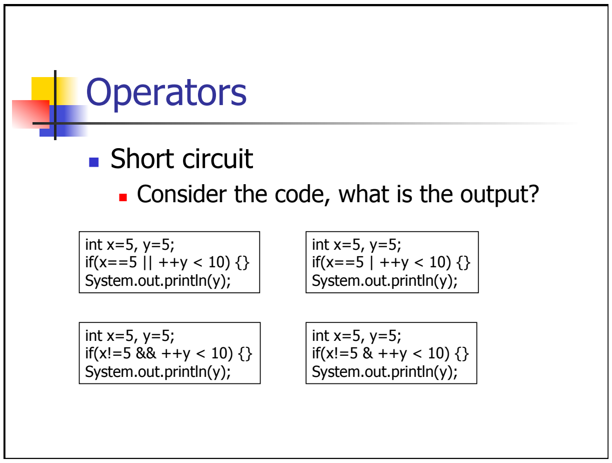
A	B	A XOR B
true	true	false
true	false	true
false	true	true
false	false	false



Operators

- Logic operators (!, &&, ||)
- Exercise:
  - **!(5==5)** returns **false**
  - **((5 == 5) && (3 > 6))** returns **false**
  - **((5 == 5) | | (3 > 6))** returns **true**

s3



Operators

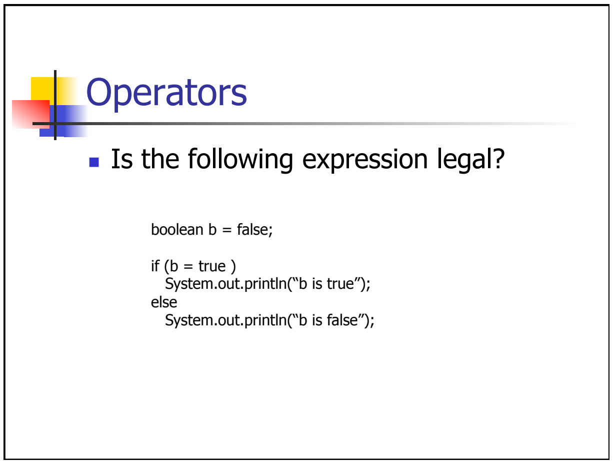
- Short circuit
  - Consider the code, what is the output?

```
int x=5, y=5;
if(x==5 || ++y < 10) {}
System.out.println(y);
```

```
int x=5, y=5;
if(x==5 | ++y < 10) {}
System.out.println(y);
```

```
int x=5, y=5;
if(x!=5 && ++y < 10) {}
System.out.println(y);
```

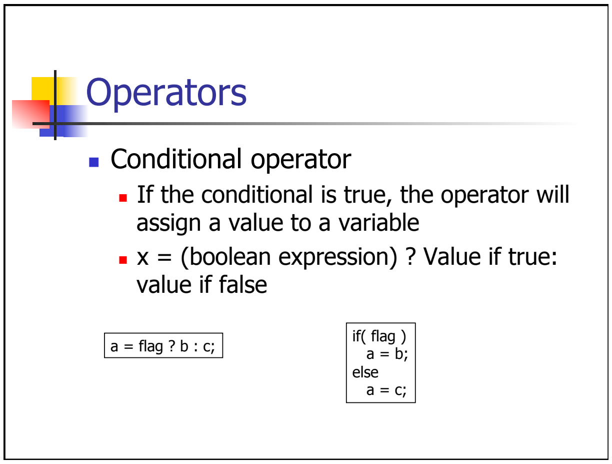
```
int x=5, y=5;
if(x!=5 & ++y < 10) {}
System.out.println(y);
```



Operators

- Is the following expression legal?

```
boolean b = false;

if (b = true )
   System.out.println(“b is true”);
else
   System.out.println(“b is false”);
```

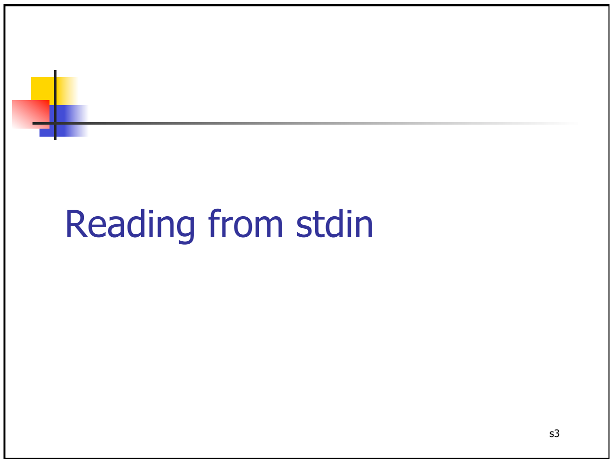


Operators

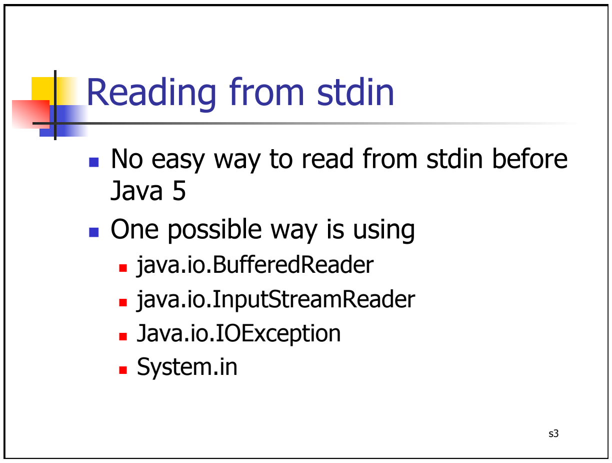
- Conditional operator
  - If the conditional is true, the operator will assign a value to a variable
  - x = (boolean expression) ? Value if true: value if false

```
a = flag ? b : c;
```

```
if( flag )
   a = b;
else
   a = c;
```

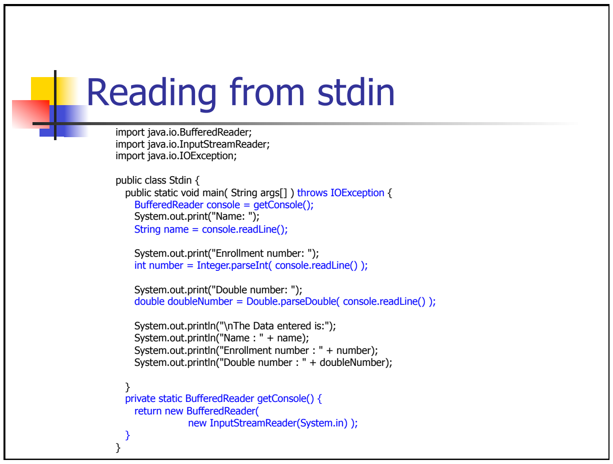


Reading from stdin



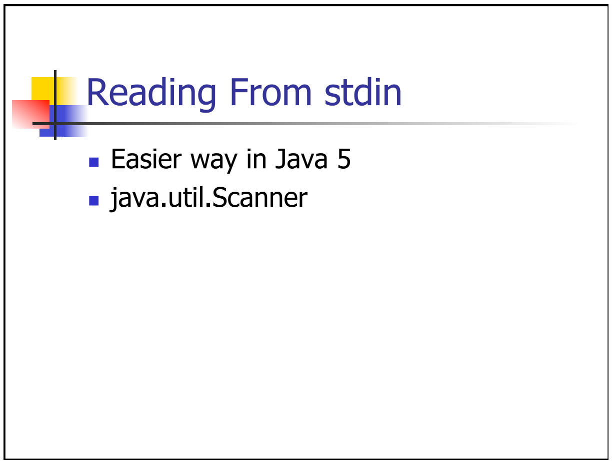
Reading from stdin

- No easy way to read from stdin before Java 5
- One possible way is using
  - java.io.BufferedReader
  - java.io.InputStreamReader
  - Java.io.IOException
  - System.in



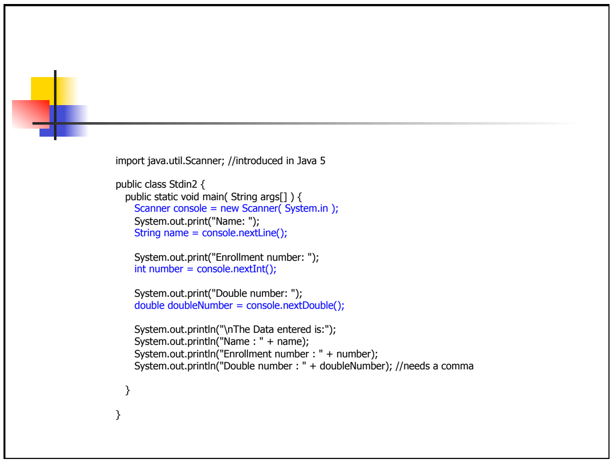
Reading from stdin

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class Stdin {
   public static void main( String args[] ) throws IOException {
      BufferedReader console = getConsole();
      System.out.print("Name: ");
      String name = console.readLine();

      System.out.print("Enrollment number: ");
      int number = Integer.parseInt( console.readLine() );

      System.out.print("Double number: ");
      double doubleNumber = Double.parseDouble( console.readLine() );

      System.out.println("\nThe Data entered is:");
      System.out.println("Name : " + name);
      System.out.println("Enrollment number : " + number);
      System.out.println("Double number : " + doubleNumber);

   }
   private static BufferedReader getConsole() {
      return new BufferedReader(
                    new InputStreamReader(System.in) );
   }
}
```

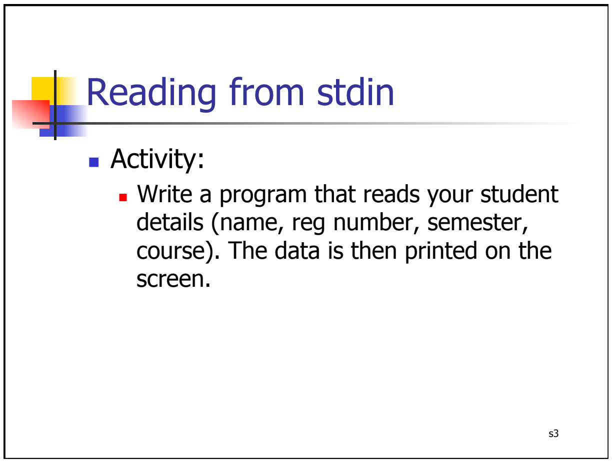


Reading From stdin

- Easier way in Java 5
- java.util.Scanner



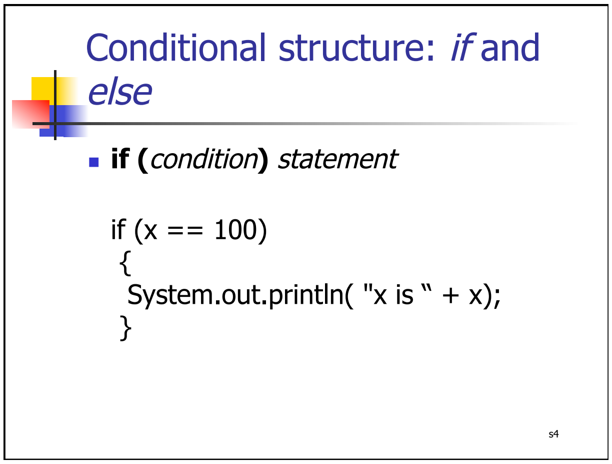
```
import java.util.Scanner; //introduced in Java 5

public class Stdin2 {
   public static void main( String args[] ) {
      Scanner console = new Scanner( System.in );
      System.out.print("Name: ");
      String name = console.nextLine();

      System.out.print("Enrollment number: ");
      int number = console.nextInt();

      System.out.print("Double number: ");
      double doubleNumber = console.nextDouble();

      System.out.println("\nThe Data entered is:");
      System.out.println("Name : " + name);
      System.out.println("Enrollment number : " + number);
      System.out.println("Double number : " + doubleNumber); //needs a comma

   }

}
```



Reading from stdin

- Activity:
  - Write a program that reads your student details (name, reg number, semester, course). The data is then printed on the screen.

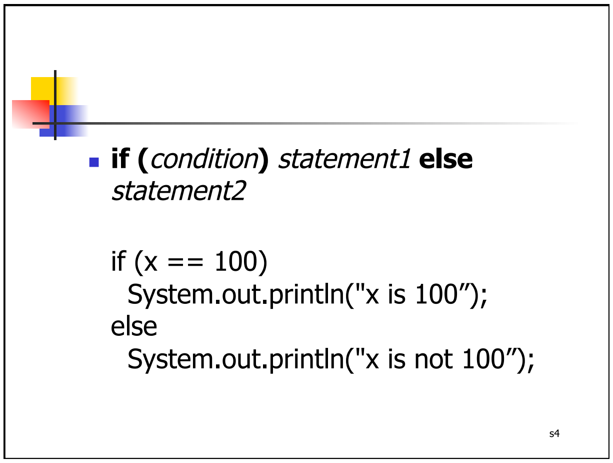
Control Structures



Conditional structure: *if* and *else*

- **if (***condition***)** *statement*

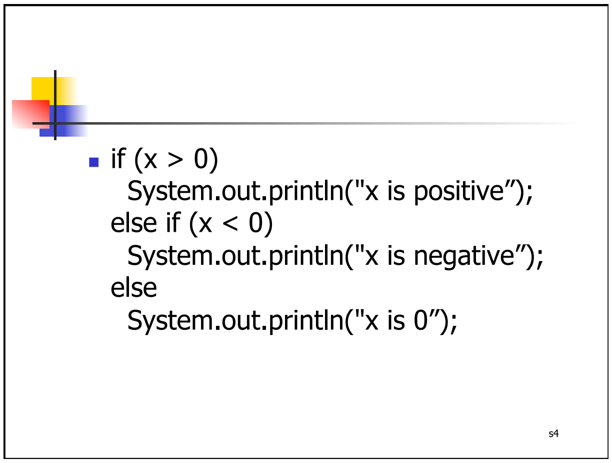
```
if (x == 100)
 {
  System.out.println( "x is �Ÿ + x);
 }
```



- **if (***condition***)** *statement1* **else** *statement2*

```
if (x == 100)
  System.out.println("x is 100”);
else
  System.out.println("x is not 100”);
```

s4



- if (x > 0)

```
if (x > 0)
  System.out.println("x is positive”);
else if (x < 0)
  System.out.println("x is negative”);
else
  System.out.println("x is 0”);
```

s4