12. Numbers

Java

Summer 2008

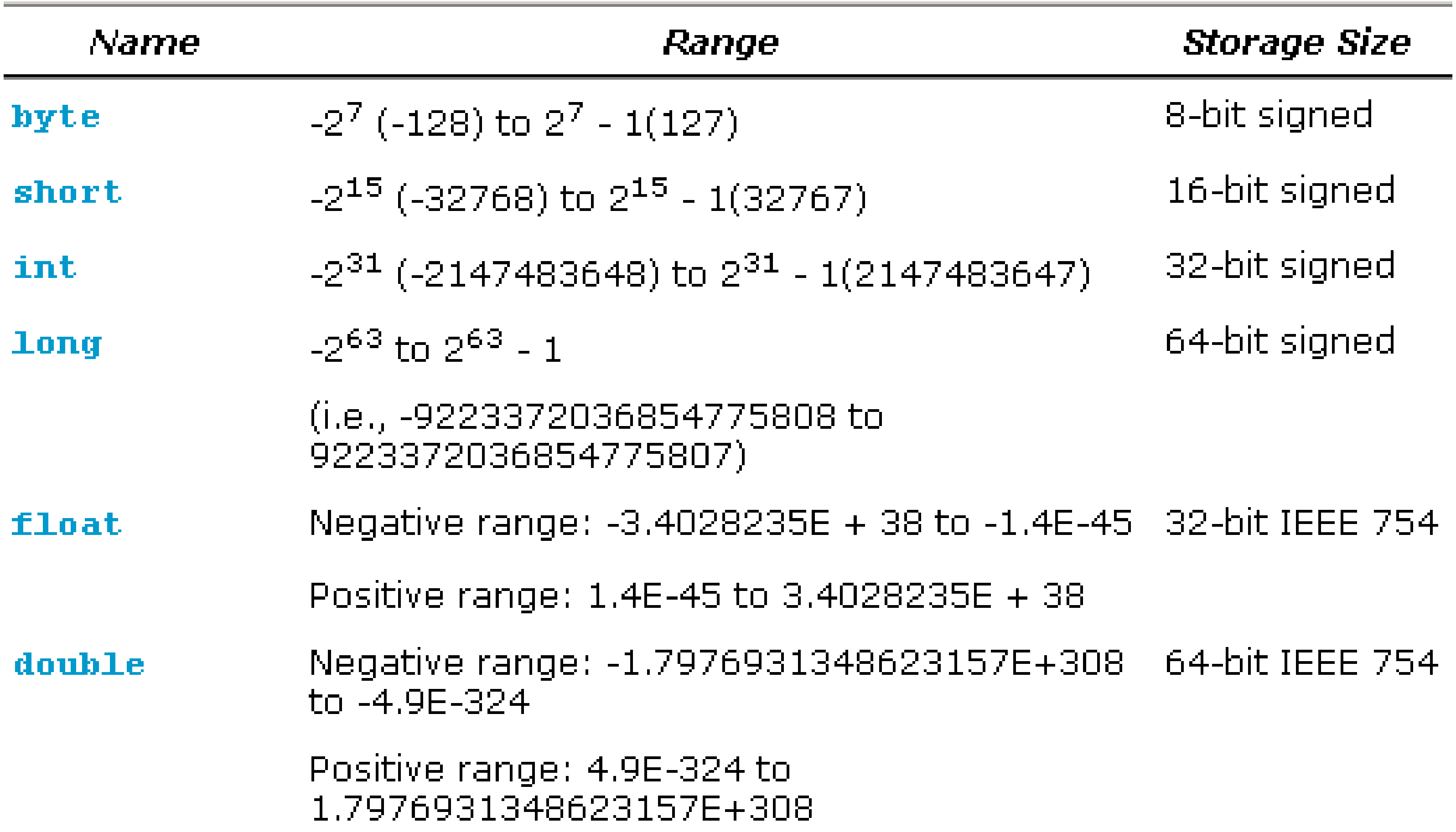
Instructor: Dr. Masoud Yaghini

Outline

- Numeric Type Conversions
- Math Class
- References

Numeric Type Conversions

Numeric Data Types (Review)



Name	*Range*	*Storage Size*
byte	-2^{7} (-128) to 2^{7} - 1(127)	8-bit signed
short	-2^{15} (-32768) to 2^{15} - 1(32767)	16-bit signed
int	-2^{31} (-2147483648) to 2^{31} - 1(2147483647)	32-bit signed
long	-2^{63} to 2^{63} - 1 (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed
float	Negative range: -3.4028235E + 38 to -1.4E-45 Positive range: 1.4E-45 to 3.4028235E + 38	32-bit IEEE 754
double	Negative range: -1.7976931348623157E+308 to -4.9E-324 Positive range: 4.9E-324 to 1.7976931348623157E+308	64-bit IEEE 754

Numeric Type Conversions

- Consider the following statements:

```
byte i = 100;
long k = i * 3 + 4;
double d = i * 3.1 + k / 2;
```

Conversion Rules

- When performing a binary operation involving two operands of different types, Java automatically converts the operand based on the following rules:

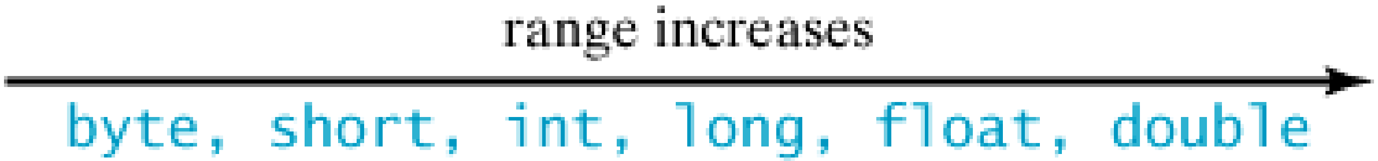
1. If one of the operands is double, the other is converted into double.
2. Otherwise, if one of the operands is float, the other is converted into float.
3. Otherwise, if one of the operands is long, the other is converted into long.
4. Otherwise, both operands are converted into int.

Numeric Type Conversions

- For example,
  - the result of 1 / 2 is 0, because both operands int values.
  - the result of 1.0 / 2 is 0.5, because 1.0 is double and 2 is converted to 2.0

Numeric Type Conversions

- You can always assign a value to a numeric variable whose type supports a larger range of values
- Thus, for instance, you can assign a long value to a float variable.



range increases

byte, short, int, long, float, double

- You cannot, however, assign a value to a variable of a type with smaller range unless you use type *casting*.

Type Casting

- ***Type casting*** is an operation that converts a value of one data type into a value of another data type.
  - ***Type widening:*** Casting a variable of a type with a small range to a variable of a type with a larger range.
  - ***Type narrowing:*** Casting a variable of a type with a large range to a variable of a type with a smaller range.

Type Casting

- Widening a type can be performed automatically.
  - `double d = 3;`
- Narrowing a type must be performed explicitly.
  - `int i = (int)3.0;`
  - `int i = (int)3.9;`
- What is wrong? int x = 5 / 2.0;

Type Casting

- Casting does not change the variable being cast.
- For example, d is not changed after casting in the following code:

```
double d = 4.5;
int i = (int)d; // d is not changed
```

Type Casting

- To assign a variable of the int type to a variable of the short or byte type, explicit casting must be used.
- For example, the following statements have a syntax error:

```
int i = 1;
byte b = i; // Error because explicit casting is required
```

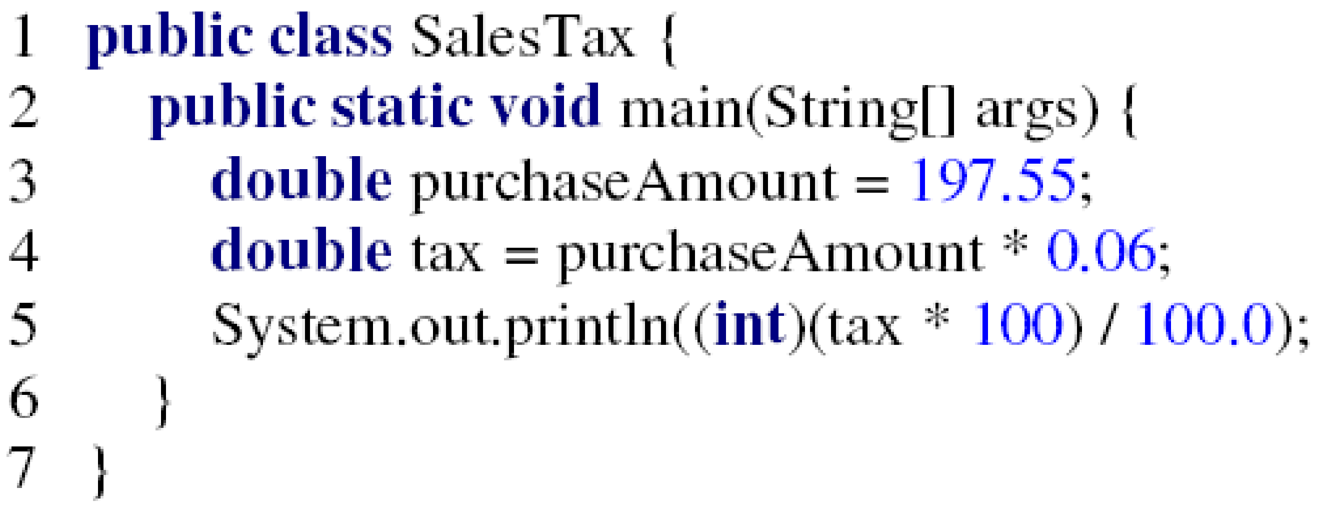
Type Casting

- Write a program that displays the sales tax with two digits after the decimal point.

```
purchaseAmount = 197.55
tax = purchaseAmount * 0.06
```

- Tax will be 11.853 , but we want the program display two digits after the decimal point.

Type Casting



```java
public class SalesTax {
    public static void main(String[] args) {
        double purchaseAmount = 197.55;
        double tax = purchaseAmount * 0.06;
        System.out.println((int)(tax * 100) / 100.0);
    }
}
```

Math Class

Math Class

- The Math class contains the methods needed to perform basic mathematical functions.
- This chapter introduces useful methods in the Math class.
- Class constants:
  - `PI (3.141...)`
  - `E  (2.718...)`

Math Class

- Math Class methods:
  - Exponent Methods
  - Rounding Methods
  - min, max, and abs
  - random Methods
  - Trigonometric Methods

Exponent Methods

- There are five methods related to exponents in the Math class:
- public static double exp(double x)
  - Return e raised to the power of x (ex)
  - Math.exp(1) returns 2.71828
- public static double log(double x)
  - Return the natural logarithm of x (ln(x) = loge(x))
  - Math.log(Math.E) returns 1.0
- public static double log10(double x)
  - Return the base 10 logarithm of x (log10(x))
  - Math.log10(10) returns 1.0

Exponent Methods

- public static double pow(double x, double b)
  - Return a raised to the power of b (xb)
  - Math.pow(2, 3) returns 8.0
  - Math.pow(3, 2) returns 9.0
  - Math.pow(3.5, 2.5) returns 22.91765
- public static double sqrt(double x)
  - Return the square root of a ()
  - Note that the parameter in the sqrt method must not be negative.
  - Math.sqrt(4) returns 2.0
  - Math.sqrt(10.5) returns 3.24

Rounding Methods

- The Math class contains five rounding methods:
- public static double ceil(double x)
  - x rounded up to its nearest integer. This integer is returned as a double value.
  - Math.ceil(2.1) returns 3.0
  - Math.ceil(2.0) returns 2.0
  - Math.ceil(–2.0) returns –2.0
  - Math.ceil(–2.1) returns -2.0
- public static double floor(double x)
  - x is rounded down to its nearest integer. This integer is returned as a double value.
  - Math.floor(2.1) returns 2.0
  - Math.floor(2.0) returns 2.0
  - Math.floor(-2.1) returns -3.0

Rounding Methods

- public static double rint(double x)
  - x is rounded to its nearest integer. If x is equally close to two integers, the even one is returned as a double.
  - Math.rint(2.1) returns 2.0
  - Math.rint(2.0) returns 2.0
  - Math.rint(3.5) returns 4.0
  - Math.rint(-2.0) returns –2.0
  - Math.rint(-2.1) returns –2.0
  - Math.rint(2.5) returns 2.0
  - Math.rint(-2.5) returns -2.0

Rounding Methods

- public static int round(float x)
  - Return (int)
  - Math.round(2.6f) returns 3 (int)
  - Math.round(–2.0f) returns -2 (int)

- public static long round(double x)
  - Return (long)
  - Math.round(2.0) returns 2 (long)
  - Math.round(–2.6) returns -3 (long)

min, max, and abs Methods

- The min and max methods are overloaded to return the minimum and maximum numbers between two numbers (int, long, float, or double).
- For example,
  - max(3.4, 5.0) returns 5.0
  - min(3, 2) returns 2
  - Math.max(2, 3) returns 3
  - Math.max(2.5, 3) returns 3.0
  - Math.min(2.5, 3.6) returns 2.5

min, max, and abs Methods

- The abs method is overloaded to return the absolute value of the number (int, long, float, and double).
- For example:
  - Math.abs(-2) returns 2
  - Math.abs(-2.1) returns 2.1

random Method

- random method generates a random double value greater than or equal to 0.0 and less than 1.0 (0 <= Math.random() < 1.0).
- You can use it to write a simple expression to generate random numbers in any range.
  - a + Math.random() * b
    - Returns a random number between a and a + b, excluding a + b.
- For example:
  - (int)(Math.random() * 10)
    - Returns a random integer between 0 and 9.
  - 50 + (int)(Math.random() * 50)
    - Returns a random integer between 50 and 99.

Trigonometric Methods

- public static double sin(double radians)
  - Math.sin(0) returns 0.0
  - Math.sin(Math.toRadians(270)) returns -1.0
  - Math.sin(Math.PI / 6) returns 0.5
  - Math.sin(Math.PI / 2) returns 1.0
- public static double cos(double radians)
  - Math.cos(0) returns 1.0
  - Math.cos(Math.PI / 6) returns 0.866
  - Math.cos(Math.PI / 2) returns 0
- public static double tan(double radians)
- public static double asin(double radians)
- public static double acos(double radians)
- public static double atan(double radians)

Trigonometric Methods

- Each method has a single double parameter, and its return type is double.
- The parameter represents an angle in radians.
- The method toRadians(double angdeg) is for converting an angle in degrees to radians
- The method toDegrees(double angrad) is for converting an angle in radians to degrees.

View java.lang.Math Documentation

- You can view the complete documentation for the Math class online from:

  http://java.sun.com/javase/6/docs/api/

References

References

- Y. Daniel Liang, **Introduction to Java Programming**, Sixth Edition, Pearson Education, 2007. (Chapter 2 & 6)

The End