



University of
Auckland

415.101

Java

Christian Collberg
August 3, 1997

Tutorial # 2

Copyright © 1997 C. Collberg

Today's Tutorial

1. Using variables in Java.
2. Printing stuff in Java
(`System.out.println`).
 - (a) Printing the basic types.
3. Arithmetic operators.
4. Use of the class `Math` for mathematical functions.
 - (a) `Random`, `sqrt`, `round`, ...
5. Input in Java.
 - (a) On the Mac.
 - (b) On the PC.
6. Choice in Java.
 - (a) Boolean expressions.
 - (b) `if`-statements.
7. Desk-Checking.

Slide 2-1

Variables in Java I

- Variables are used to store information.
- The information can come from many sources:
 1. It is often provided by the user. (It is **read** by the program.)
 2. It can be **temporary** data that we need to store to solve a particular problem.
- Variables have **type**. The type of a variable describes what kind of data we can store in it. A **string** variable can only store strings, an **int** variable only integers, a **double** variable only real numbers, etc.

Slide 2-2

Variables in Java II

- We must **declare** a variable before we can use it. The declaration gives the **name** of the variable and its **type**:

```
int hello;  
double bob;  
String message;
```
- We use **assignment** statements to store data into a variable:

```
hello = 5;  
bob = 5.5;  
message = "Hello Dolly!"
```
- We can also initialize a variable directly in the declaration:

```
int hello = 5;  
String message = "Hello Dolly!"
```

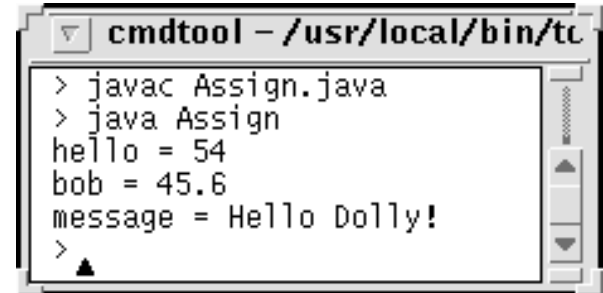
Slide 2-3

Variables in Java III

- Variable values can be used:
`System.out.println(hello);`
`bob = bob + 5.5;`
- Only certain types of assignments are legal:
 - `ints` can be assigned to `ints`, `doubles` to `doubles`, `Strings` to `Strings`, etc.
 - `ints` can be assigned to `doubles`; the integer is automatically converted to a real number.
- Variables must be **initialized** (given an initial value) before they can be used. These statements are wrong:
`int hello, bob;`
`System.out.println(hello);`
`bob = bob + 5.5;`

Slide 2-4

```
class Assign {  
    public static void main (String [] args) {  
        int hello;  
        double bob;  
        String message;  
  
        hello = 54;  
        bob = 45.6;  
        message = "Hello Dolly!";  
  
        System.out.println("hello = " + hello);  
        System.out.println("bob = " + bob);  
        System.out.println("message = "+message);  
    }  
}
```



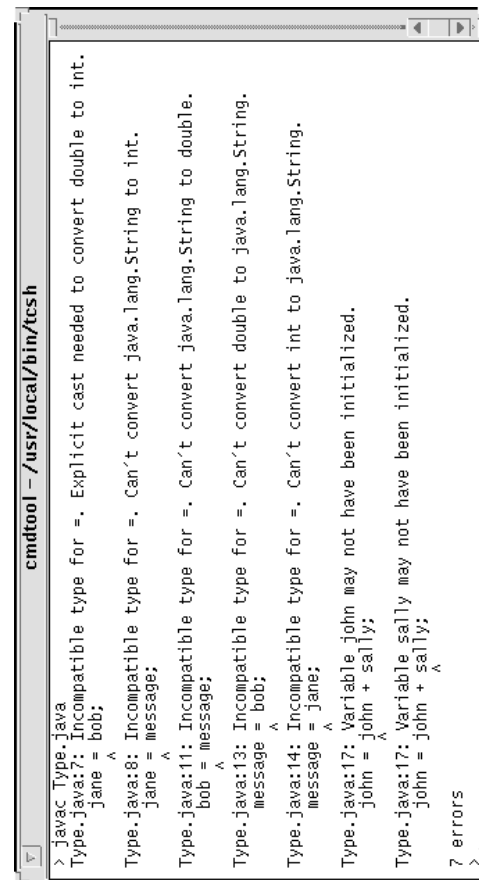
```
cmdtool - /usr/local/bin/tc  
> javac Assign.java  
> java Assign  
hello = 54  
bob = 45.6  
message = Hello Dolly!  
>
```

Slide 2-5

- What does the compiler print when given this program?

```
class Type {  
    public static void main (String [] args) {  
        int jane      = 5;  
        double bob     = 45.6;  
        String message = "Jane Dolly!";  
  
        jane = bob;  
        jane = message;  
  
        bob = jane;  
        bob = message;  
  
        message = bob;  
        message = jane;  
  
        int john, sally;  
        john = john + sally;  
  
        int karl = 54;  
        karl = karl + 56;  
    }  
}
```

Slide 2-6



```
cmdtool - /usr/local/bin/tcsh  
> javac Type.java  
Type.java:7: Incompatible type for =. Explicit cast needed to convert double to int.  
    jane = bob;  
        ^  
Type.java:8: Incompatible type for =. Can't convert java.lang.String to int.  
    jane = message;  
        ^  
Type.java:11: Incompatible type for =. Can't convert java.lang.String to double.  
    bob = message;  
        ^  
Type.java:13: Incompatible type for =. Can't convert double to java.lang.String.  
    message = bob;  
            ^  
Type.java:14: Incompatible type for =. Can't convert int to java.lang.String.  
    message = jane;  
            ^  
Type.java:17: Variable john may not have been initialized.  
    john = john + sally;  
        ^  
Type.java:17: Variable sally may not have been initialized.  
    john = john + sally;  
        ^  
7 errors  
>
```

Slide 2-7

```

public class java.io.PrintStream
    extends java.io.FilterOutputStream {
    public void flush();

    public void print(boolean b);
    public void print(char c);
    public void print(char s[]);
    public void print(double d);
    public void print(float f);
    public void print(int i);
    public void print(long l);
    public void print(Object obj);
    public void print(String s);
    public void println();

    public void println(boolean b);
    public void println(char c);
    public void println(char s[]);
    public void println(double d);
    public void println(float f);
    public void println(int i);
    public void println(long l);
    public void println(Object obj);
    public void println(String s);
}

```

Slide 2-8

Arithmetic Operations

- Java has the standard built-in arithmetic operators:

$a + b$ Add a and b .

$a - b$ Subtract a and b .

$a * b$ Multiply a and b .

a / b Divide a and b , i.e. $\frac{a}{b}$.

$a \% b$ Remainder when a is divided by b , i.e.

$$a \% b = a - b * \lfloor \frac{a}{b} \rfloor.$$

- If both a and b are integers then the result of $a \oplus b$ is an integer ($\oplus$ is one of $+$, $-$, $*$, $/$).
- If a or b (or both) are real numbers then the result of $a \oplus b$ is a real number.
- What will the output be from the program in the next slide?

Slide 2-9

```

class Arith {
    public static void main (String [] args) {
        int a = 23;
        int b = 2;
        double x = 23.4;
        double y = 2.4;

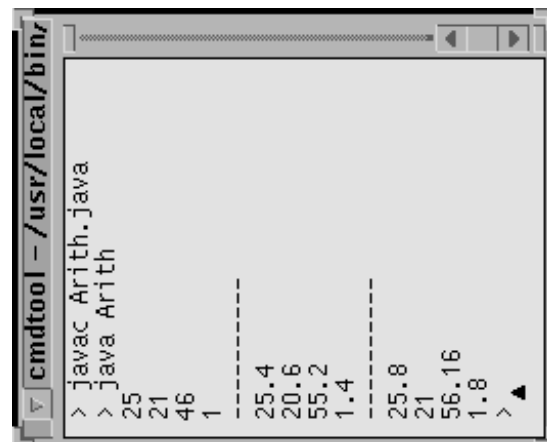
        System.out.println(a + b);
        System.out.println(a - b);
        System.out.println(a * b);
        System.out.println(a % b);
        System.out.println("-----");

        System.out.println(a + y);
        System.out.println(a - y);
        System.out.println(a * y);
        System.out.println(a % y);
        System.out.println("-----");

        System.out.println(x + y);
        System.out.println(x - y);
        System.out.println(x * y);
        System.out.println(x % y);
    }
}

```

Slide 2-10



```

System.out.println(a + b);
System.out.println(a - b);
System.out.println(a * b);
System.out.println(a % b);

System.out.println(a + y);
System.out.println(a - y);
System.out.println(a * y);
System.out.println(a % y);

System.out.println(x + y);
System.out.println(x - y);
System.out.println(x * y);
System.out.println(x % y);

```

Slide 2-11

The Class Math

- Java has most common mathematical operators (+,*,/,-, etc.) built into the language.
- Other standard mathematical functions (trigonometric, exponentiation, etc) are supplied by a standard, built-in class called Math.

```
public final class java.lang.Math
    extends java.lang.Object {
    public final static double E=...;
    public final static double PI=...;

    public static int max(int a, int b);
    public static int min(int a, int b);
    public static int round(float a);

    public static double abs(double a);

    public static double ceil(double a);
    public static double floor(double a);
```

Slide 2-12

```
public static double exp(double a);
public static double log(double a);
public static double pow(double a,
                        double b);
public static double sqrt(double a);

public static double max(double a,
                        double b);
public static double min(double a,
                        double b);
public static double random();

public static double acos(double a);
public static double asin(double a);
public static double atan(double a);
public static double atan2(double a,
                        double b);
public static double cos(double a);
public static double sin(double a);
public static double tan(double a);
}
```

Slide 2-13

Class Math Problem #1—#2

1. Write a program which “shows” that

$$x^y \equiv e^{y \cdot \ln x}$$

by trying the formula for some values of x and y .

2. Do the same thing, but this time choose x and y to be random numbers between 5.0 and 20.0. (NOTE: `Math.random()` returns a double between 0.0 and 1.0.)

```
cmdtool - /usr/local/bin/tcsh
> javac Exp.java
> java Exp
Math.exp(4.78 * Math.log(5.67)) = 4000.62
Math.pow(5.67, 4.78) = 4000.62

Math.exp(7.79 * Math.log(23.1)) = 4.19312e+10
Math.pow(23.1, 7.79) = 4.19312e+10
>
```

Slide 2-14

```
class Exp {
    public static void main (String [] args) {
        double x; double y;
        x = 5.67; y = 4.78;
        System.out.println(
            "Math.exp(" + y + " * Math.log(" + x + ")) = " +
            Math.exp(y * Math.log(x)));
        System.out.println(
            "Math.pow(" + x + ", " + y + ") = " + Math.pow(x,y) + "\n");
        x = 23.1; y = 7.79;
        System.out.println(
            "Math.exp(" + y + " * Math.log(" + x + ")) = " +
            Math.exp(y * Math.log(x)));
        System.out.println(
            "Math.pow(" + x + ", " + y + ") = " + Math.pow(x,y));
    }
}
```

Slide 2-15

Input in Java

```
class Exp {
    public static void main (String [] args) {
        double x = 5.0 + 10.0*Math.random();
        double y = 5.0 + 10.0*Math.random();

        System.out.println(
            "Math.exp(" + y + " * Math.log(" + x + ")") = " +
            Math.exp(y * Math.log(x));
        System.out.println(
            "Math.pow(" + x + " , " + y + ") = " + Math.pow(x,y));
    }
}
```

Slide 2-16

- Most programs interact with the user in some way. We have already seen how printing to the screen using `System.out.println` allows us the program to provide information **to** the user.
- You will be provided with two classes called **Input**. Which one to use depends on which system you use.
- **Input** provides methods for getting information **from** the user.
- `Input.toText("Prompt")` will print **Prompt** on the screen (or put up a window with "Prompt" in it) and allow the user to enter a text (string). That string will then be returned to the program.

Slide 2-17

- These are the operations provided by the class **Input**. There are operations for reading strings, integers, and reals. If you don't provide a prompt, a *default* prompt will be given.

```
public class Input {
    public static String toText(String prompt)
    {...}
    public static String toText() {
        return toText("Enter a string");
    }

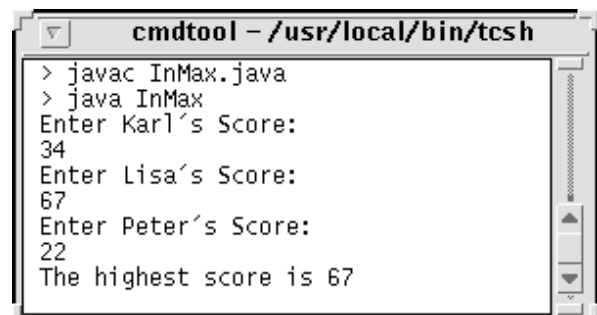
    public static int toInt(String prompt)
    {...}
    public static int toInt() {
        return toInt("Enter an integer");
    }

    public static double toDouble(String prompt)
    {...}
    public static double toDouble() {
        return toDouble("Enter a double");
    }
}
```

Slide 2-18

Class Input Problem

1. Karl, Lisa, and Peter are playing a game and want a program to help to figure out who's winning. Write a program that reads Karl's, Lisa's, and Peter's scores and writes the highest score.



Slide 2-19

```

class InMax {
    public static void main (String [] args) {
        int KarlsScore = Input.toInt ("Enter Karl's Score:");
        int LisasScore = Input.toInt ("Enter Lisa's Score:");
        int PetersScore = Input.toInt ("Enter Peter's Score:");

        int MaxScore = Math.max(KarlsScore,
                                Math.max(LisasScore, PetersScore));

        System.out.println("The highest score is " + MaxScore);
    }
}

```

Slide 2–20

Choice in Java

- Real programs make *choices*. Depending on some condition, we either do one thing, or another. Such conditions are called **Boolean Expressions**.
- In Java we write:


```

if (Condition)
    DoThis;

```
- We can compare numbers to numbers and strings to strings, but not numbers to strings. What's the result of the program fragment below?

```

int    i = 15;
String s = "Hello";
double d = 15.5;
if (i==15)    System.out.println("i is 15");
if (d==i)     System.out.println("d is 15");
if (s=="Bye") System.out.println("s is Bye");
if (s==i)     System.out.println("s is 15");

```

Slide 2–21

Choice Problem

Write a program which

1. reads in the names of two tri-athletes, and their three individual scores,
2. Computes and writes out the two total scores and the name of the winner, if any.

```

cmdtool - /usr/local/bin/tcsh
> javac IfMax.java
> java fMax
Enter the 1st contestant's name:
Bart
Enter the 1st contestant's 3 scores:
10
7
13
Enter the 2nd contestant's name:
Lisa
Enter the 2nd contestant's 3 scores:
11
21
15
Bart scored 30
Lisa scored 47
The highest score is 47
Lisa wins!
>

```

Slide 2–22

```

public static void main (String [] args) {
    /*A*/ String FirstContestant = Input.toText ("Enter the 1st [...] name:");
    /*B*/ int FirstContestantScore = Input.toInt ("Enter the [...] 3 scores:");
    /*C*/ FirstContestantScore = FirstContestantScore + Input.toInt ("");
    /*D*/ FirstContestantScore = FirstContestantScore + Input.toInt ("");

    /*E*/ String SecondContestant = Input.toText ("Enter the 2nd [...] name:");
    /*F*/ int SecondContestantScore = Input.toInt ("Enter the 2nd [...] 3 scores:");
    /*G*/ SecondContestantScore = SecondContestantScore + Input.toInt ("");
    /*H*/ SecondContestantScore = SecondContestantScore + Input.toInt ("");

    /*I*/ System.out.println(FirstContestant + " scored " + FirstContestantScore);
    /*J*/ System.out.println(SecondContestant + " scored " + SecondContestantScore);
    /*K*/ System.out.println("The highest score is " +
        Math.max(FirstContestantScore, SecondContestantScore));

    /*L*/ String Winner = "Noone";
    /*M*/ if (FirstContestantScore>SecondContestantScore) Winner=FirstContestant;
    /*N*/ if (FirstContestantScore<SecondContestantScore) Winner=SecondContestant;
    /*O*/ System.out.println("\n" + Winner + " wins!");
}

```

Slide 2–23

Desk-Checking

*Eskimos have 500 words for snow.
Computer Scientists have 500 words
for failure.*

- When programs don't work we must find out what's gone wrong.
- We can "desk-check" the program, i.e. play like we're a computer.
- Make a table with one row for every statement in the program. Make one column for every variable.
- Step through the program, one statement at a time, and fill in the table as the variables change.
- Desk-check the program in the previous slide!

Slide 2–24

Program Point	Program Variables				Winner
	First-Contestant	FirstCon-testantScore	Second-Contestant	SecondCon-testantScore	
/*A*/	"Bart"				
Output="Enter the 1st contestant's name:"					
/*B*/	"Bart"	10			
Output="Enter the 1st contestant's 3 scores:"					
/*C*/	"Bart"	17			
/*D*/	"Bart"	30			
/*E*/	"Bart"	30	"Lisa"		
Output="Enter the 2nd contestant's name:"					
/*F*/	"Bart"	30	"Lisa"	11	
/*G*/	"Bart"	30	"Lisa"	32	

Slide 2–25

Program Point	Program Variables				Winner
	First-Contestant	FirstCon-testantScore	Second-Contestant	SecondCon-testantScore	
/*H*/	"Bart"	30	"Lisa"	47	
/*I*/	"Bart"	30	"Lisa"	47	
/*J*/	"Bart"	30	"Lisa"	47	
/*K*/	"Bart"	30	"Lisa"	47	
Output="The highest score is 47"					
/*L*/	"Bart"	30	"Lisa"	47	"Noone"
/*M*/	"Bart"	30	"Lisa"	47	"Noone"
/*N*/	"Bart"	30	"Lisa"	47	"Lisa"
/*O*/	"Bart"	30	"Lisa"	47	"Lisa"
Output="Lisa wins!"					

Slide 2–26