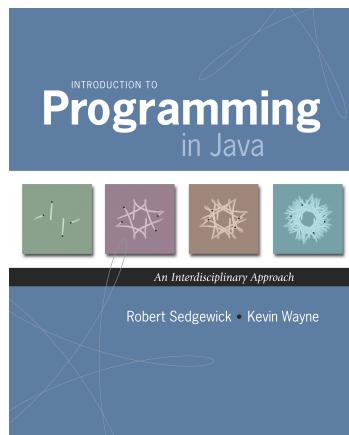


1.1 Your First Program



1

Languages

Machine languages. Tedious and error-prone.

Natural languages. Ambiguous; can be difficult to parse.

High-level programming languages. Acceptable tradeoff.

"Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do." – Donald Knuth



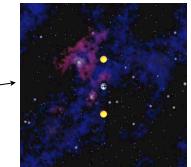
3

Why Programming?

Why programming? Need to tell computer what you want it to do.

Naive ideal. Natural language instructions.

"Please simulate the motion of these heavenly bodies, subject to Newton's laws of motion and gravity."



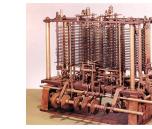
Prepackaged solutions (apps)? Great, when what they do is what you want.



Programming. Enables you to make a computer do **anything** you want.



Ada Lovelace



Analytic Engin

well, almost anything
[stay tuned]

1

Why Program?

Why program?

- A natural, satisfying and creative experience.
 - Enables accomplishments not otherwise possible
 - Opens new world of intellectual endeavor.

First challenge. Learn a programming language.

Next question. Which one?



Naive ideal. A single programming language.

3

Our Choice: Java

Java features.

- Widely used.
- Widely available.
- Embraces full set of modern abstractions.
- Variety of automatic checks for mistakes in programs.

Java economy. ← \$100 billion,
5 million developers



James Gosling
<http://java.net/jog>

- Mars rover.
- Cell phones.
- Blu-ray Disc.
- Web servers.
- Medical devices.
- Supercomputing.
- ...

5

Why Java?

Java features.

- Widely used.
- Widely available.
- Embraces full set of modern abstractions.
- Variety of automatic checks for mistakes in programs.

Facts of life.

- No language is perfect.
- We need to choose **some** language.

"There are only two kinds of programming languages: those people always [gripe] about and those nobody uses."

— Bjarne Stroustrup



Our approach.

- Minimal subset of Java.
- Develop general programming skills that are applicable to many languages

It's not about the language!

6

A Rich Subset of the Java Language

Built-In Types	
int	double
long	String
char	boolean

System	
System.out.println()	
System.out.print()	
System.out.printf()	

Math Library	
Math.sin()	Math.cos()
Math.log()	Math.exp()
Math.sqrt()	Math.pow()
Math.min()	Math.max()
Math.abs()	Math.PI

Flow Control	
if	else
for	while

Parsing	
Integer.parseInt()	
Double.parseDouble()	

Primitive Numeric Types		
+	-	*
/	%	++
--	>	<
<=	>=	==
!=		

Boolean	
true	false
	&&
!	

Punctuation	
{	}
()
,	;

Assignment	
=	

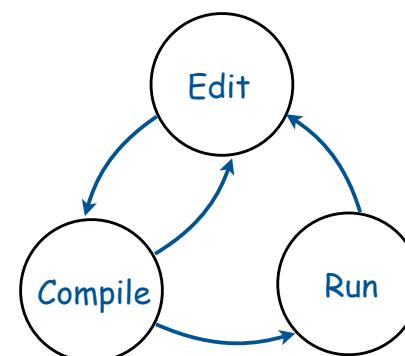
String	
+	""
length()	compareTo()
charAt()	matches()

Arrays	
a[i]	
new	
a.length	

Objects	
class	static
public	private
final	toString()
new	main()

7

Program Development



8

Programming in Java

Programming in Java.

- **Create** the program by typing it into a text editor, and save it as `HelloWorld.java`.

```
*****  
 * Prints "Hello, World"  
 * Everyone's first Java program.  
*****  
  
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

`HelloWorld.java`

9

Programming in Java

Programming in Java.

- Create the program by typing it into a text editor, and save it as `HelloWorld.java`.
- **Compile** it by typing at the command-line:
`javac HelloWorld.java`

command-line → `% javac HelloWorld.java`

- This creates a Java bytecode file named: `HelloWorld.class`.

10

Programming in Java

Programming in Java.

- Create the program by typing it into a text editor, and save it as `HelloWorld.java`.
- **Compile** it by typing at the command-line:
`javac HelloWorld.java`.
- **Execute** it by typing at the command-line:
`java HelloWorld`.

command-line → `% javac HelloWorld.java`

```
% java HelloWorld  
Hello, World
```

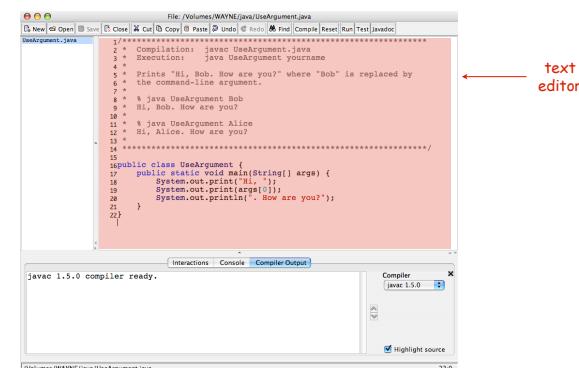
11

Program Development (using DrJava)

Program development in Java (using DrJava).



1. **Edit** your program using the built-in text editor.
2. **Compile** it to create an executable file.
3. Run your program.

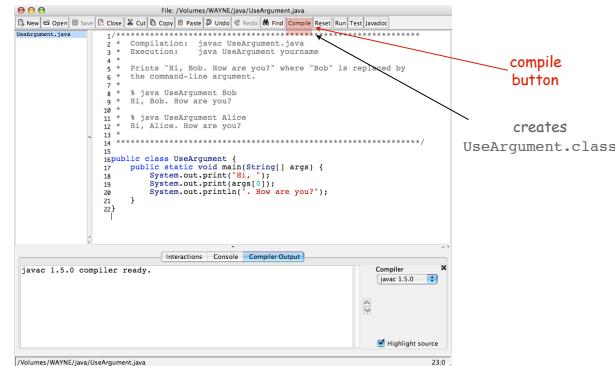


12

Program Development (using DrJava)

Program development in Java (using DrJava).

1. Edit your program.
2. **Compile** it by clicking the "compile" button.
3. Run your program.



The screenshot shows the DrJava IDE interface. The code editor contains a Java file named `UseArgument.java` with the following content:

```

/*
 * Compilation: javac UseArgument.java
 * Execution: java UseArgument [username]
 * Prints "Hi, [username]. How are you?" where "[username]" is replaced by
 * the command-line argument.
 */
public class UseArgument {
    public static void main(String[] args) {
        System.out.print("Hi, ");
        System.out.print(args[0]);
        System.out.println(", How are you?");
    }
}

```

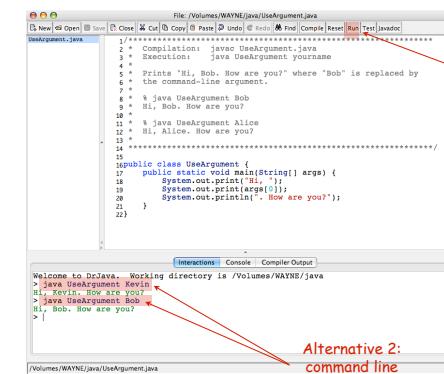
A red arrow points from the word "compile" in the list above to the "Compile" button in the toolbar. Another red arrow points from the text "creates UseArgument.class" to the "Compiler Output" tab, which displays the message "javac 1.5.0 compiler ready."

13

Program Development (using DrJava)

Program development in Java (using DrJava).

1. Edit your program.
2. Compile it to create an executable file.
3. **Run** your program by clicking the "run" button or using the command line.



The screenshot shows the DrJava IDE interface. The code editor contains a Java file named `UseArgument.java` with the same code as before. A red arrow points from the text "Alternative 1: run button (OK if no args)" to the "Run" button in the toolbar. The "Compiler Output" tab shows the message "javac 1.5.0 compiler ready." The "Interactions" tab shows the following session:

```

Welcome to DrJava. Working directory is /Volumes/WAYNE/java
> java UseArgument Kevin
Hi, Kevin. How are you?
> java UseArgument Bob
Hi, Bob. How are you?
> java UseArgument Alice
Hi, Alice. How are you?
>

```

Alternative 1:
run button
(OK if no args)

both use
UseArgument.class

Alternative 2:
command line
(to provide args)

14

Note: Program Style

Three versions of the same program.



The screenshot shows the DrJava IDE interface with three versions of a Java program. Each version includes a portrait of a person.

Version 1:

```

// java HelloWorld
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World");
    }
}

```

Version 2:

```

/*
 * Compilation: javac HelloWorld.java
 * Execution: java HelloWorld
 * Prints "Hello, World". By tradition, this is everyone's first program.
 */
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}

```

Version 3:

```

public class HelloWorld { public static void main(String[] args) { System.out.println("Hello, World"); } }

```

Annotations on the left side of the code blocks indicate that fonts, color, comments, and extra space are not relevant to Java.

15

Note: Program Style

Emphasizing consistent style can

- Make it easier to spot errors.
- Make it easier for others to read and use code.
- Enable development environment to provide useful visual cues.

Bottom line for COS 126:

- Let the Doctor indent for you.
- Correct any style problems automatically discovered when you submit.
- Follow your preceptor/grader's advice on style.

16

1.2 Built-in Types of Data



17

Built-in Data Types

Data type. A set of values and operations defined on those values.

type	set of values	literal values	operations
char	characters	'A' '@'	compare
String	sequences of characters	"Hello World" "CS is fun"	concatenate
int	integers	17 12345	add, subtract, multiply, divide
double	floating-point numbers	3.1415 6.022e23	add, subtract, multiply, divide
boolean	truth values	true false	and, or, not

18

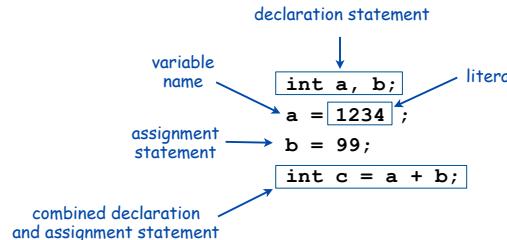
Basic Definitions

Variable. A name that refers to a value.

Literal. Programming-language representation of a value.

Assignment statement. Associates a value with a variable.

Program. Sequence of statements.



19

Trace

Trace. Table of variable values after each statement.

	a	b	t
int a, b;	undefined	undefined	undefined
a = 1234;	1234	undefined	undefined
b = 99;	1234	99	undefined
int t = a;	1234	99	1234
a = b;	99	99	1234
b = t;	99	1234	1234

20

Text

String data type. Useful for program input and output.

values	sequences of characters
typical literals	"Hello, " "1" " " * "
operation	concatenate
operator	+

string data type

Important note: meaning of characters depends on context!

"1234" + " " + " " + "99"
 ↑ ↑ ↑
 character operator operator

expression	value
"Hi, " + "Bob"	"Hi, Bob"
"1" + " 2 " + "1"	"1 2 1"
"1234" + " " + "99"	"1234 + 99"
"1234" + "99"	"123499"

String concatenation examples

"1234" + " " + "99"
 ↑ ↑ ↑
 white space space characters white space

21

Example: Subdivisions of a Ruler

```
public class Ruler
{
    public static void main(String[] args)
    {
        String ruler1 = "1";
        String ruler2 = ruler1 + " 2 " + ruler1;
        String ruler3 = ruler2 + " 3 " + ruler2;
        String ruler4 = ruler3 + " 4 " + ruler3;
        System.out.println(ruler4);
    }
}
```

"1"
 "1 2 1"
 "1 2 1 3 1 2 1"

```
% java Ruler
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
```

string concatenation



22

Integers

int data type. Useful for calculations, expressing algorithms.

values	integers between -2^{31} and $+2^{31} - 1$				
typical literals	1234 99 -99 0 1000000				
operations	add subtract multiply divide remainder				
operators	+	-	*	/	%

int data type

there is a largest int and a smallest int

expression	value	comment
5 + 3	8	
5 - 3	2	
5 * 3	15	
5 / 3	1	no fractional part
5 % 3	2	remainder
1 / 0		run-time error
3 * 5 - 2	13	* has precedence
3 + 5 / 2	5	/ has precedence
3 - 5 - 2	-4	left associative
(3 - 5) - 2	-4	better style

examples of int operations

23

Integer Operations

```
public class IntOps
{
    public static void main(String[] args)
    {
        int a = Integer.parseInt(args[0]); ← command-line arguments
        int b = Integer.parseInt(args[1]);
        int sum = a + b;
        int prod = a * b;
        int quot = a / b;
        int rem = a % b;
        System.out.println(a + " + " + b + " = " + sum);
        System.out.println(a + " * " + b + " = " + prod);
        System.out.println(a + " / " + b + " = " + quot);
        System.out.println(a + " % " + b + " = " + rem);
    }
}
```

Java automatically converts a, b, and rem to type String

1234 = 12*99 + 46

```
% javac IntOps.java
% java IntOps 1234 99
1234 + 99 = 1333
1234 * 99 = 122166
1234 / 99 = 12
1234 % 99 = 46
```

24

Floating-Point Numbers

double data type. Useful in scientific applications.

values	approximations to real numbers				
typical literals	3.14159	6.022e23	-3.0	2.0	1.4142135623730951
operations	add +	subtract -	multiply *	divide /	remainder %
double data type					
expression		value			
3.141 + .03	3.171				
3.141 - .03	3.111				
6.02e23/2	3.01E+23				
5.0 / 3.0	1.666666666666700				
10.0 % 3.141	0.577				
1.0 / 0.0	Infinity	special value			
Math.sqrt(2.0)	1.414213562373100				
Math.sqrt(-1.0)	NaN	special value "not a number"			
examples of double operations					

25

Excerpts from Java's Math Library

public class Math	
double abs(double a)	absolute value of a
double max(double a, double b)	maximum of a and b
double min(double a, double b)	minimum of a and b
	also defined for int, long, and float
double sin(double theta)	sine function
double cos(double theta)	cosine function
double tan(double theta)	tangent function
	In radians. Use toDegrees() and toRadians() to convert.
double exp(double a)	exponential (e^a)
double log(double a)	natural log ($\log_e a$, or $\ln a$)
double pow(double a, double b)	raise a to the bth power (a^b)
long round(double a)	found to the nearest integer
double random()	random number in [0, 1)
double sqrt(double a)	square root of a
double E	value of e (constant)
double PI	value of p (constant)

26

Quadratic Equation

Ex. Solve quadratic equation $x^2 + bx + c = 0$.

$$\text{roots} = \frac{-b \pm \sqrt{b^2 - 4c}}{2}$$

```
public class Quadratic
{
    public static void main(String[] args)
    {
        // Parse coefficients from command-line.
        double b = Double.parseDouble(args[0]);
        double c = Double.parseDouble(args[1]);

        // Calculate roots.
        double discriminant = b*b - 4.0*c;
        double d = Math.sqrt(discriminant);
        double root1 = (-b + d) / 2.0;
        double root2 = (-b - d) / 2.0;

        // Print them out.
        System.out.println(root1);
        System.out.println(root2);
    }
}
```

27

Testing

Testing. Some valid and invalid inputs.

```
% java Quadratic -3.0 2.0
2.0
1.0
command-line arguments
```

```
% java Quadratic -1.0 -1.0
1.618033988749895
-0.6180339887498949
golden ratio
```

```
% java Quadratic 1.0 1.0
NaN
NaN
"not a number"
```

```
% java Quadratic 1.0 hello
java.lang.NumberFormatException: hello
```

```
% java Quadratic 1.0
java.lang.ArrayIndexOutOfBoundsException
```

$x^2 - 3x + 2$

$x^2 - x - 1$

$x^2 + x + 1$

28

Booleans

boolean data type. Useful to control logic and flow of a program.

values literals	true or false
operations operators	true false
and or not	&& !
	boolean data type

a	!a	a	b	a && b	a b
true	false	false	false	false	false
false	true	false	true	false	true
		true	false	false	true
		true	true	true	true

Truth-table definitions of boolean operations

Comparison Operators

Comparison operators.

- Two operands of the same type.
- Result: a value of type **boolean**.

op	meaning	true	false
==	equal	2 == 2	2 == 3
!=	not equal	3 != 2	2 != 2
<	less than	2 < 13	2 < 2
<=	less than or equal	2 <= 2	3 <= 2
>	greater than	13 > 2	2 > 13
>=	greater than or equal	3 >= 2	2 >= 3

comparison operators

non-negative discriminant? $(b * b - 4.0 * a * c) \geq 0.0$
beginning of a century? $(year \% 100) == 0$
legal month? $(month \geq 1) \&\& (month \leq 12)$

comparison examples

Leap Year

Q. Is a given year a leap year?

A. Yes if either (i) divisible by 400 or (ii) divisible by 4 but not 100.

```
public class LeapYear
{
    public static void main(String[] args)
    {
        int year = Integer.parseInt(args[0]);
        boolean isLeapYear;

        // divisible by 4 but not 100
        isLeapYear = (year % 4 == 0) && (year % 100 != 0);

        // or divisible by 400
        isLeapYear = isLeapYear || (year % 400 == 0);

        System.out.println(isLeapYear);
    }
}
```

```
% java LeapYear 2004
true

% java LeapYear 1900
false

% java LeapYear 2000
true
```

Type Conversion

Type conversion. Convert from one type of data to another.

- Automatic (done by Java when no loss of precision; or with strings).
- Explicitly defined by function call.
- Cast (write desired type within parens).

expression	type	value	
"1234" + 99	String	"123499"	automatic
Integer.parseInt("123")	int	123	explicit
(int) 2.71828	int	2	cast
Math.round(2.71828)	long	3	explicit
(int) Math.round(2.71828)	int	3	cast
(int) Math.round(3.14159)	int	3	cast
11 * 0.3	double	3.3	automatic
(int) 11 * 0.3	double	3.3	cast, automatic
11 * (int) 0.3	int	0	cast
(int) (11 * 0.3)	int	3	cast, automatic



Pay attention to the type of your data.

← type conversion can give counterintuitive results
but gets easier to understand with practice

Type Conversion Example: Random Integer

Ex. Generate a pseudo-random number between 0 and N-1.

```
public class RandomInt
{
    public static void main(String[] args)
    {
        int N = Integer.parseInt(args[0]);
        double r = Math.random();           ← String to int (method)
        int n = (int) (r * N);             ← double between 0.0 and 1.0
                                         ← double to int (cast)   ← int to double (automatic)
                                         ← int to String (automatic)

        System.out.println("random integer is " + n);
    }
}

% java RandomInt 6
random integer is 3

% java RandomInt 6
random integer is 0

% java RandomInt 10000
random integer is 3184
```

33

Summary

A **data type** is a set of values and operations on those values.

- **String** text processing, input and output.
- **double, int** mathematical calculation.
- **boolean** decision making.



Example of bad type conversion

Be aware. In Java you must:

- Declare type of values.
- Convert between types when necessary.

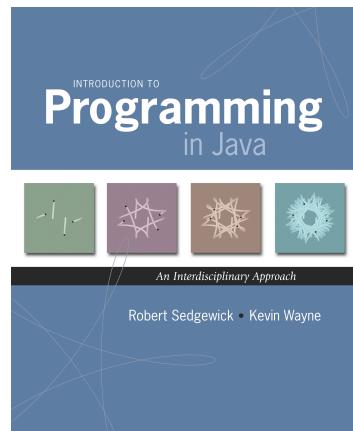
Why do we need types?

- Type conversion must be done at some level.
- Compiler can help do it correctly.
- Example: In 1996, Ariane 5 rocket exploded after takeoff because of bad type conversion.



34

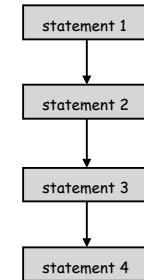
1.3 Conditionals and Loops



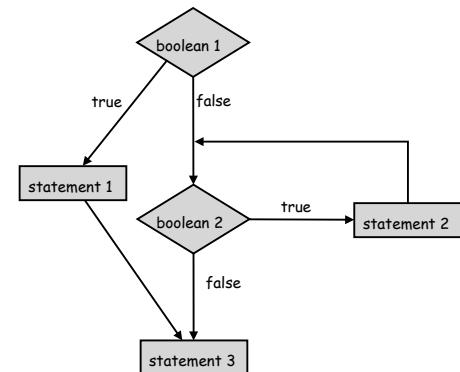
Control Flow

Control flow.

- Sequence of statements that are actually executed in a program.
- Conditionals and loops: enable us to choreograph control flow.



straight-line control flow



control flow with conditionals and loops

Conditionals



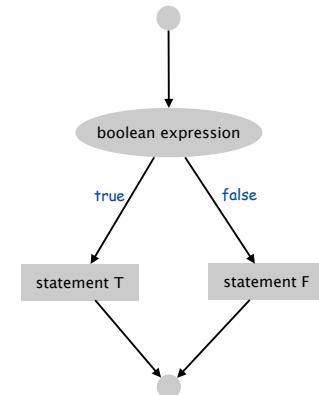
37

If Statement

The **if statement**. A common branching structure.

- Evaluate a boolean expression.
- If true, execute some statements.
- If false, execute other statements.

```
if (boolean expression) {  
    statement T;  
}  
else {  
    statement F; // can be any sequence  
    of statements  
}
```

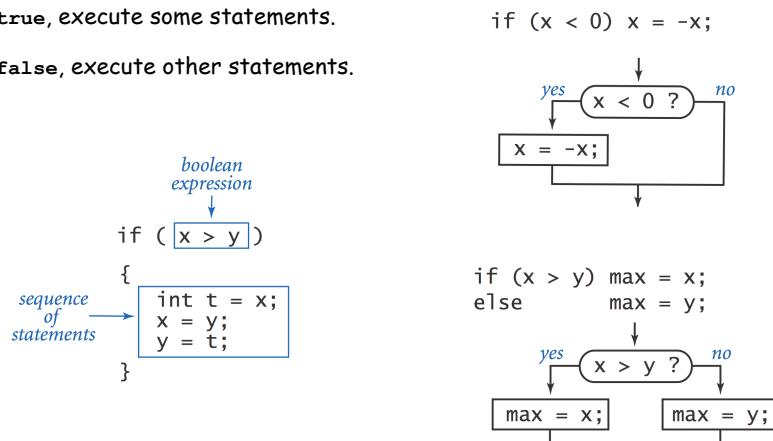


38

If Statement

The **if statement**. A common branching structure.

- Evaluate a boolean expression.
- If true, execute some statements.
- If false, execute other statements.

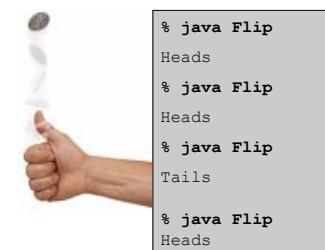


39

If Statement

Ex. Take different action depending on value of variable.

```
public class Flip {  
    public static void main(String[] args) {  
        if (Math.random() < 0.5) System.out.println("Heads");  
        else System.out.println("Tails");  
    }  
}
```



```
% java Flip  
Heads  
% java Flip  
Heads  
% java Flip  
Tails  
% java Flip  
Heads
```

40

If-Else: Leap Year revisited

If-else. Take different action depending on value of variable.

- If `isLeapYear` is true, then print "is a".
 - Otherwise, print "isn't a".

```
System.out.print(year + " ");

if (isLeapYear) {
    System.out.print("is a");
}
else {
    System.out.print("isn't a");
}

System.out.println(" leap year")
```

If-Else: Leap Year revisited

```
public class LeapYear
{
    public static void main(String[] args)
    {
        int year = Integer.parseInt(args[0]);
        boolean isLeapYear;

        // divisible by 4 but not 100
        isLeapYear = (year % 4 == 0) && (year % 100 != 0);

        // or divisible by 400
        isLeapYear = isLeapYear || (year % 400 == 0);

        System.out.print(year + " ");

        if (isLeapYear) {
            System.out.print("is a");
        }
        else {
            System.out.print("isn't a");
        }

        System.out.println(" leap year");
    }
}
```

% java LeapYear
2004 is a leap year

% java LeapYear
1900 isn't a leap year

% java LeapYear
2000 is a leap year

```
% java LeapYear 2004  
2004 is a leap year  
  
% java LeapYear 1900  
1900 isn't a leap year  
  
% java LeapYear 2000  
2000 is a leap year
```

Oblivious Sorting

Sort. Read in 3 integers and rearrange them in ascending order.

```
public class Sort3 {  
    public static void main(String[] args) {  
  
        int a = Integer.parseInt(args[0]);  
        int b = Integer.parseInt(args[1]);  
        int c = Integer.parseInt(args[2]);  
  
        if (b > c) { int t = b; b = c; c = t; } ← swap b and c  
        if (a > b) { int t = a; a = b; b = t; } ← swap a and b  
        if (b > c) { int t = b; b = c; c = t; } ← swap b and c  
  
        System.out.println(a + " " + b + " " + c);  
    }  
}
```

% java Sort3 9 8
7 8 9

% java Sort3 2 1
1 2 7

Puzzle 1. Sort 4 integers with 5 compare-exchanges.

Puzzle 2. Sort 6 integers with 12.

The Hello World Collection

The latest additions are shown in **bold**.

1 [JC-Enterprise](#)

4 [4test](#)

A

[ABAAP4](#) ActionScript [ActionScript 3.0](#) ActionScript-Flash8 [Actionscript-Flash8](#) ActionScript-Flash8R [Actionscript-Flash8X](#) Ada [AdaPL](#) [Algol-60](#) [Algol-68](#) [Algol-68X](#) [Amber-ES](#) [AMOS](#) [APC](#) [Arc](#) [AppleScript](#) [Arena](#) [Asp](#) [ASP-C](#) [ASP-JavaScript](#) [ASP-JS](#) [ASP-VBS](#) [ASP-.NET](#) [Assembler-6502-AppleII](#) [Assembler-6502-C64](#) [Assembler-6502-Commodore64](#) [Assembler-6502-AstariST](#) [Assembler-66008](#) [Assembler-ARM](#) [Assembler-Darwin-PFC](#) [Assembler-HP3000](#) [Assembler-HLA](#) [Assembler-IBM-370](#) [Assembler-Itanium](#) [Assembler-MIPS](#) [Assembler-MINIX](#) [Assembler-MOOC](#) [Assembler-PDP11](#) [Assembler-PDP10](#) [Assembler-PDP15](#) [Assembler-Win32](#) [Assembler-Z80-Console](#) [Assembler-Z80I](#) [Autotkey](#) [AutoIT](#) [AutoIt3](#) [awk](#) [Axel](#)

B [BAL](#) [BASIC-PCANS BASIC-RCB](#) BeanShell [Befunge](#) [Beta](#) [BjDB](#) [BitTNT](#) [BitZilla](#) [BlitzBasic](#) [Blitz3D](#) [BMC Remedy](#) [Boo Brainfuck](#) [BS2000](#) [BuddyScript](#) [ByteCode](#)

C [C](#) [C++/C/C++/C++-Boost](#) [C++-FITS](#) [C++-GML](#) [C++-ISO](#) [C++-Qt](#) [C/C++](#) [C-Amiga/Anywhere](#) [C-ANSI](#) [C-Curses](#) [C-GEM](#) [C-Ibutton](#) [C-KiB](#) [C-Objective C](#) [C-OpenGLES](#) [C-Perl](#) [C-PostgreSQL](#) [C-SQLite](#) [C-XMOS](#) [Cylinder](#) [C-XML](#) [C-XML-RPC](#)

D [D](#) [Darkerhouse](#) [Database](#) [Datafile](#) [dbs](#) [DCL](#) [Dialect](#) [Dialect DM](#) [DMX](#) [Dylan](#) [Dynamorphism](#)

E [E](#) [exam](#) [Eiffel](#) [Elan](#) [Elastic](#) [Europa](#)

F [F](#) [False](#) [Ferde](#) [Flamek](#) [Flapin](#) [Flaming Thunder](#) [Focus](#) [Forth](#) [Forthian](#) [Forthian77](#) [Forthian90](#) [Forthian99](#) [Fortress](#) [FreeBASIC](#) [Frink](#)

G [G](#) [Go](#) [Gambas](#) [GameMonkey](#) [Script](#) [Genie](#) [aimagine](#) [Gentoo](#) [GLBasic](#) [GM](#) [Gofer](#) [GoogleGadgets](#) [GRAMophone](#) [Groovy](#) [GykoSoft](#)

H [Haskell](#) [HDK](#) [HDL](#) [HDL2](#) [HDL3](#) [HDL4](#) [HDL5](#) [Human](#) [HyperTalk](#)

I [I](#) [IMB-Ever2](#) [ICL-ICSL](#) [Icon](#) [ID3](#) [Inform](#) [Inform-4GL](#) [Ingris-ABF](#) [InstallScript](#) [Interval](#) [Io](#) [Isarcare](#)

J [J](#) [Jade](#) [Java](#) [Java-Mobile](#) [Java-Server-Pages](#) [Java-Servlet](#) [Java-Swing](#) [JavaScript](#) [JCL](#) [Jesse](#) [JavaScript](#)

K [K](#) [Kotlin](#)

L [L](#) [Lahey](#)

M [M](#) [MacOS](#) [MacOSX](#) [Mame](#) [Mame-0.144](#) [Mame-0.145](#) [Mame-0.146](#) [Mame-0.147](#) [Mame-0.148](#) [Mame-0.149](#) [Mame-0.150](#) [Mame-0.151](#) [Mame-0.152](#) [Mame-0.153](#) [Mame-0.154](#) [Mame-0.155](#) [Mame-0.156](#) [Mame-0.157](#) [Mame-0.158](#) [Mame-0.159](#) [Mame-0.160](#) [Mame-0.161](#) [Mame-0.162](#) [Mame-0.163](#) [Mame-0.164](#) [Mame-0.165](#) [Mame-0.166](#) [Mame-0.167](#) [Mame-0.168](#) [Mame-0.169](#) [Mame-0.170](#) [Mame-0.171](#) [Mame-0.172](#) [Mame-0.173](#) [Mame-0.174](#) [Mame-0.175](#) [Mame-0.176](#) [Mame-0.177](#) [Mame-0.178](#) [Mame-0.179](#) [Mame-0.180](#) [Mame-0.181](#) [Mame-0.182](#) [Mame-0.183](#) [Mame-0.184](#) [Mame-0.185](#) [Mame-0.186](#) [Mame-0.187](#) [Mame-0.188](#) [Mame-0.189](#) [Mame-0.190](#) [Mame-0.191](#) [Mame-0.192](#) [Mame-0.193](#) [Mame-0.194](#) [Mame-0.195](#) [Mame-0.196](#) [Mame-0.197](#) [Mame-0.198](#) [Mame-0.199](#) [Mame-0.200](#) [Mame-0.201](#) [Mame-0.202](#) [Mame-0.203](#) [Mame-0.204](#) [Mame-0.205](#) [Mame-0.206](#) [Mame-0.207](#) [Mame-0.208](#) [Mame-0.209](#) [Mame-0.210](#) [Mame-0.211](#) [Mame-0.212](#) [Mame-0.213](#) [Mame-0.214](#) [Mame-0.215](#) [Mame-0.216](#) [Mame-0.217](#) [Mame-0.218](#) [Mame-0.219](#) [Mame-0.220](#) [Mame-0.221](#) [Mame-0.222](#) [Mame-0.223](#) [Mame-0.224](#) [Mame-0.225](#) [Mame-0.226](#) [Mame-0.227](#) [Mame-0.228](#) [Mame-0.229](#) [Mame-0.230](#) [Mame-0.231](#) [Mame-0.232](#) [Mame-0.233](#) [Mame-0.234](#) [Mame-0.235](#) [Mame-0.236](#) [Mame-0.237](#) [Mame-0.238](#) [Mame-0.239](#) [Mame-0.240](#) [Mame-0.241](#) [Mame-0.242](#) [Mame-0.243](#) [Mame-0.244](#) [Mame-0.245](#) [Mame-0.246](#) [Mame-0.247](#) [Mame-0.248](#) [Mame-0.249](#) [Mame-0.250](#) [Mame-0.251](#) [Mame-0.252](#) [Mame-0.253](#) [Mame-0.254](#) [Mame-0.255](#) [Mame-0.256](#) [Mame-0.257](#) [Mame-0.258](#) [Mame-0.259](#) [Mame-0.260](#) [Mame-0.261](#) [Mame-0.262](#) [Mame-0.263](#) [Mame-0.264](#) [Mame-0.265](#) [Mame-0.266](#) [Mame-0.267](#) [Mame-0.268](#) [Mame-0.269](#) [Mame-0.270](#) [Mame-0.271](#) [Mame-0.272](#) [Mame-0.273](#) [Mame-0.274](#) [Mame-0.275](#) [Mame-0.276](#) [Mame-0.277](#) [Mame-0.278](#) [Mame-0.279](#) [Mame-0.280](#) [Mame-0.281](#) [Mame-0.282](#) [Mame-0.283](#) [Mame-0.284](#) [Mame-0.285](#) [Mame-0.286](#) [Mame-0.287](#) [Mame-0.288](#) [Mame-0.289](#) [Mame-0.290](#) [Mame-0.291](#) [Mame-0.292](#) [Mame-0.293](#) [Mame-0.294](#) [Mame-0.295](#) [Mame-0.296](#) [Mame-0.297](#) [Mame-0.298](#) [Mame-0.299](#) [Mame-0.300](#) [Mame-0.301](#) [Mame-0.302](#) [Mame-0.303](#) [Mame-0.304](#) [Mame-0.305](#) [Mame-0.306](#) [Mame-0.307](#) [Mame-0.308](#) [Mame-0.309](#) [Mame-0.310](#) [Mame-0.311](#) [Mame-0.312](#) [Mame-0.313](#) [Mame-0.314](#) [Mame-0.315](#) [Mame-0.316](#) [Mame-0.317](#) [Mame-0.318](#) [Mame-0.319](#) [Mame-0.320](#) [Mame-0.321](#) [Mame-0.322](#) [Mame-0.323](#) [Mame-0.324](#) [Mame-0.325](#) [Mame-0.326](#) [Mame-0.327](#) [Mame-0.328](#) [Mame-0.329](#) [Mame-0.330](#) [Mame-0.331](#) [Mame-0.332](#) [Mame-0.333](#) [Mame-0.334](#) [Mame-0.335](#) [Mame-0.336](#) [Mame-0.337](#) [Mame-0.338](#) [Mame-0.339](#) [Mame-0.340](#) [Mame-0.341](#) [Mame-0.342](#) [Mame-0.343](#) [Mame-0.344](#) [Mame-0.345](#) [Mame-0.346](#) [Mame-0.347](#) [Mame-0.348](#) [Mame-0.349](#) [Mame-0.350](#) [Mame-0.351](#) [Mame-0.352](#) [Mame-0.353](#) [Mame-0.354](#) [Mame-0.355](#) [Mame-0.356](#) [Mame-0.357](#) [Mame-0.358](#) [Mame-0.359](#) [Mame-0.360](#) [Mame-0.361](#) [Mame-0.362](#) [Mame-0.363](#) [Mame-0.364](#) [Mame-0.365](#) [Mame-0.366](#) [Mame-0.367](#) [Mame-0.368](#) [Mame-0.369](#) [Mame-0.370](#) [Mame-0.371](#) [Mame-0.372](#) [Mame-0.373](#) [Mame-0.374](#) [Mame-0.375](#) [Mame-0.376](#) [Mame-0.377](#) [Mame-0.378](#) [Mame-0.379](#) [Mame-0.380](#) [Mame-0.381](#) [Mame-0.382](#) [Mame-0.383](#) [Mame-0.384](#) [Mame-0.385](#) [Mame-0.386](#) [Mame-0.387](#) [Mame-0.388](#) [Mame-0.389](#) [Mame-0.390](#) [Mame-0.391](#) [Mame-0.392](#) [Mame-0.393](#) [Mame-0.394](#) [Mame-0.395](#) [Mame-0.396](#) [Mame-0.397](#) [Mame-0.398](#) [Mame-0.399](#) [Mame-0.400](#) [Mame-0.401](#) [Mame-0.402](#) [Mame-0.403](#) [Mame-0.404](#) [Mame-0.405](#) [Mame-0.406](#) [Mame-0.407](#) [Mame-0.408](#) [Mame-0.409](#) [Mame-0.410](#) [Mame-0.411](#) [Mame-0.412](#) [Mame-0.413](#) [Mame-0.414](#) [Mame-0.415](#) [Mame-0.416](#) [Mame-0.417](#) [Mame-0.418](#) [Mame-0.419](#) [Mame-0.420](#) [Mame-0.421](#) [Mame-0.422](#) [Mame-0.423](#) [Mame-0.424](#) [Mame-0.425](#) [Mame-0.426](#) [Mame-0.427](#) [Mame-0.428](#) [Mame-0.429](#) [Mame-0.430](#) [Mame-0.431](#) [Mame-0.432](#) [Mame-0.433](#) [Mame-0.434](#) [Mame-0.435](#) [Mame-0.436](#) [Mame-0.437](#) [Mame-0.438](#) [Mame-0.439](#) [Mame-0.440](#) [Mame-0.441](#) [Mame-0.442](#) [Mame-0.443](#) [Mame-0.444](#) [Mame-0.445](#) [Mame-0.446](#) [Mame-0.447](#) [Mame-0.448](#) [Mame-0.449](#) [Mame-0.450](#) [Mame-0.451](#) [Mame-0.452](#) [Mame-0.453](#) [Mame-0.454](#) [Mame-0.455](#) [Mame-0.456](#) [Mame-0.457](#) [Mame-0.458](#) [Mame-0.459](#) [Mame-0.460](#) [Mame-0.461](#) [Mame-0.462](#) [Mame-0.463](#) [Mame-0.464](#) [Mame-0.465](#) [Mame-0.466](#) [Mame-0.467](#) [Mame-0.468](#) [Mame-0.469](#) [Mame-0.470](#) [Mame-0.471](#) [Mame-0.472](#) [Mame-0.473](#) [Mame-0.474](#) [Mame-0.475](#) [Mame-0.476](#) [Mame-0.477](#) [Mame-0.478](#) [Mame-0.479](#) [Mame-0.480](#) [Mame-0.481](#) [Mame-0.482](#) [Mame-0.483](#) [Mame-0.484](#) [Mame-0.485](#) [Mame-0.486](#) [Mame-0.487](#) [Mame-0.488](#) [Mame-0.489](#) [Mame-0.490](#) [Mame-0.491](#) [Mame-0.492](#) [Mame-0.493](#) [Mame-0.494](#) [Mame-0.495](#) [Mame-0.496](#) [Mame-0.497](#) [Mame-0.498](#) [Mame-0.499](#) [Mame-0.500](#) [Mame-0.501](#) [Mame-0.502](#) [Mame-0.503](#) [Mame-0.504](#) [Mame-0.505](#) [Mame-0.506](#) [Mame-0.507](#) [Mame-0.508](#) [Mame-0.509](#) [Mame-0.510](#) [Mame-0.511](#) [Mame-0.512](#) [Mame-0.513](#) [Mame-0.514](#) [Mame-0.515](#) [Mame-0.516](#) [Mame-0.517](#) [Mame-0.518](#) [Mame-0.519](#) [Mame-0.520](#) [Mame-0.521](#) [Mame-0.522](#) [Mame-0.523](#) [Mame-0.524](#) [Mame-0.525](#) [Mame-0.526](#) [Mame-0.527](#) [Mame-0.528](#) [Mame-0.529](#) [Mame-0.530](#) [Mame-0.531](#) [Mame-0.532](#) [Mame-0.533](#) [Mame-0.534](#) [Mame-0.535](#) [Mame-0.536](#) [Mame-0.537](#) [Mame-0.538](#) [Mame-0.539](#) [Mame-0.540](#) [Mame-0.541](#) [Mame-0.542](#) [Mame-0.543](#) [Mame-0.544](#) [Mame-0.545](#) [Mame-0.546](#) [Mame-0.547](#) [Mame-0.548](#) [Mame-0.549](#) [Mame-0.550](#) [Mame-0.551](#) [Mame-0.552](#) [Mame-0.553](#) [Mame-0.554](#) [Mame-0.555](#) [Mame-0.556](#) [Mame-0.557](#) [Mame-0.558](#) [Mame-0.559](#) [Mame-0.560](#) [Mame-0.561](#) [Mame-0.562](#) [Mame-0.563](#) [Mame-0.564](#) [Mame-0.565](#) [Mame-0.566](#) [Mame-0.567](#) [Mame-0.568](#) [Mame-0.569](#) [Mame-0.570](#) [Mame-0.571](#) [Mame-0.572](#) [Mame-0.573](#) [Mame-0.574](#) [Mame-0.575](#) [Mame-0.576](#) [Mame-0.577](#) [Mame-0.578](#) [Mame-0.579](#) [Mame-0.580](#) [Mame-0.581](#) [Mame-0.582](#) [Mame-0.583](#) [Mame-0.584](#) [Mame-0.585](#) [Mame-0.586](#) [Mame-0.587](#) [Mame-0.588](#) [Mame-0.589](#) [Mame-0.590](#) [Mame-0.591](#) [Mame-0.592](#) [Mame-0.593](#) [Mame-0.594](#) [Mame-0.595](#) [Mame-0.596](#) [Mame-0.597](#) [Mame-0.598](#) [Mame-0.599](#) [Mame-0.600](#) [Mame-0.601](#) [Mame-0.602](#) [Mame-0.603](#) [Mame-0.604](#) [Mame-0.605](#) [Mame-0.606](#) [Mame-0.607](#) [Mame-0.608](#) [Mame-0.609](#) [Mame-0.610](#) [Mame-0.611](#) [Mame-0.612](#) [Mame-0.613](#) [Mame-0.614](#) [Mame-0.615](#) [Mame-0.616](#) [Mame-0.617](#) [Mame-0.618](#) [Mame-0.619](#) [Mame-0.620](#) [Mame-0.621](#) [Mame-0.622](#) [Mame-0.623](#) [Mame-0.624](#) [Mame-0.625](#) [Mame-0.626](#) [Mame-0.627](#) [Mame-0.628](#) [Mame-0.629](#) [Mame-0.630](#) [Mame-0.631](#) [Mame-0.632](#) [Mame-0.633](#) [Mame-0.634](#) [Mame-0.635](#) [Mame-0.636](#) [Mame-0.637](#) [Mame-0.638](#) [Mame-0.639](#) [Mame-0.640](#) [Mame-0.641](#) [Mame-0.642](#) [Mame-0.643](#) [Mame-0.644](#) [Mame-0.645](#) [Mame-0.646](#) [Mame-0.647](#) [Mame-0.648](#) [Mame-0.649](#) [Mame-0.650](#) [Mame-0.651](#) [Mame-0.652](#) [Mame-0.653](#) [Mame-0.654](#) [Mame-0.655](#) [Mame-0.656](#) [Mame-0.657](#) [Mame-0.658](#) [Mame-0.659](#) [Mame-0.660](#) [Mame-0.661](#) [Mame-0.662](#) [Mame-0.663](#) [Mame-0.664](#) [Mame-0.665](#) [Mame-0.666](#) [Mame-0.667](#) [Mame-0.668](#) [Mame-0.669](#) [Mame-0.670](#) [Mame-0.671](#) [Mame-0.672](#) [Mame-0.673](#) [Mame-0.674](#) [Mame-0.675](#) [Mame-0.676](#) [Mame-0.677](#) [Mame-0.678](#) [Mame-0.679](#) [Mame-0.680](#) [Mame-0.681](#) [Mame-0.682](#) [Mame-0.683](#) [Mame-0.684](#) [Mame-0.685](#) [Mame-0.686](#) [Mame-0.687](#) [Mame-0.688](#) [Mame-0.689](#) [Mame-0.690](#) [Mame-0.691](#) [Mame-0.692](#) [Mame-0.693](#) [Mame-0.694](#) [Mame-0.695](#) [Mame-0.696](#) [Mame-0.697](#) [Mame-0.698](#) [Mame-0.699](#) [Mame-0.700](#) [Mame-0.701](#) [Mame-0.702](#) [Mame-0.703](#) [Mame-0.704](#) [Mame-0.705](#) [Mame-0.706](#) [Mame-0.707](#) [Mame-0.708](#) [Mame-0.709](#) [Mame-0.710](#) [Mame-0.711](#) [Mame-0.712](#) [Mame-0.713](#) [Mame-0.714](#) [Mame-0.715](#) [Mame-0.716](#) [Mame-0.717](#) [Mame-0.718](#) [Mame-0.719](#) [Mame-0.720](#) [Mame-0.721](#) [Mame-0.722](#) [Mame-0.723](#) [Mame-0.724](#) [Mame-0.725](#) [Mame-0.726](#) [Mame-0.727](#) [Mame-0.728](#) [Mame-0.729](#) [Mame-0.730](#) [Mame-0.731](#) [Mame-0.732](#) [Mame-0.733](#) [Mame-0.734](#) [Mame-0.735](#) [Mame-0.736](#) [Mame-0.737](#) [Mame-0.738](#) [Mame-0.739](#) [Mame-0.740](#) [Mame-0.741](#) [Mame-0.742](#) [Mame-0.743](#) [Mame-0.744](#) [Mame-0.745](#) [Mame-0.746](#) [Mame-0.747](#) [Mame-0.748](#) [Mame-0.749](#) [Mame-0.750](#) [Mame-0.751](#) [Mame-0.752](#) [Mame-0.753](#) [Mame-0.754](#) [Mame-0.755](#) [Mame-0.756](#) [Mame-0.757](#) [Mame-0.758](#) [Mame-0.759](#) [Mame-0.760](#) [Mame-0.761](#) [Mame-0.762](#) [Mame-0.763](#) [Mame-0.764](#) [Mame-0.765](#) [Mame-0.766](#) [Mame-0.767](#) [Mame-0.768](#) [Mame-0.769](#) [Mame-0.770](#) [Mame-0.771](#) [Mame-0.772](#) [Mame-0.773](#) [Mame-0.774](#) [Mame-0.775](#) [Mame-0.776](#) [Mame-0.777](#) [Mame-0.778](#) [Mame-0.779](#) [Mame-0.780](#) [Mame-0.781](#) [Mame-0.782](#) [Mame-0.783](#) [Mame-0.784](#) [Mame-0.785](#) [Mame-0.786](#) [Mame-0.787](#) [Mame-0.788](#) [Mame-0.789](#) [Mame-0.790](#) [Mame-0.791](#) [Mame-0.792](#) [Mame-0.793](#) [Mame-0.794](#) [Mame-0.795](#) [Mame-0.796](#) [Mame-0.797](#) [Mame-0.798](#) [Mame-0.799](#) [Mame-0.800](#) [Mame-0.801](#) [Mame-0.802](#) [Mame-0.803](#) [Mame-0.804](#) [Mame-0.805](#) [Mame-0.806](#) [Mame-0.807](#) [Mame-0.808](#) [Mame-0.809](#) [Mame-0.810](#) [Mame-0.811](#) [Mame-0.812](#) [Mame-0.813](#) [Mame-0.814](#) [Mame-0.815](#) [Mame-0.816](#) [Mame-0.817](#) [Mame-0.818](#) [Mame-0.819](#) [Mame-0.820](#) [Mame-0.821](#) [Mame-0.822](#) [Mame-0.823](#) [Mame-0.824](#) [Mame-0.825](#) [Mame-0.826](#) [Mame-0.827](#) [Mame-0.828](#) [Mame-0.829](#) [Mame-0.830](#) [Mame-0.831](#) [Mame-0.832](#) [Mame-0.833](#) [Mame-0.834](#) [Mame-0.835](#) [Mame-0.836](#) [Mame-0.837](#) [Mame-0.838](#) [Mame-0.839](#) [Mame-0.840</](#)