Java Language Basics

COSC 122 Programming II

Rick Halterman
Valley View University
June 2012

Variables

- Local variables: declare and use just like in C/C++
  - Java has fewer primitive types
  - Unlike C/C++, Java's primitive types are exactly specified; e.g., an **`int`** is a 32-bit two's complement signed value
  - Floating-point values follow IEEE 754 floating-point standard

Primitive Data Types

Name	Meaning	Range	Size
`byte`	byte	$-128 \ldots +127$	8 bits
`short`	short integer	$-32,768 \ldots +32,767$	16 bits
`char`	Unicode character	$0 \ldots +65,536$	16 bits
`int`	integer	$-2,147,483,648 \ldots +2,147,483,647$	32 bits
`long`	long integer	$-9,223,372,036,854,775,808$ $\ldots +9,223,372,036,854,775,807$	64 bits
`float`	single-precision floating point	$\pm 3.4 \times 10^{+38} \ldots \pm 1.4 \times 10^{-45}$ with at least 7 decimal digits of precision	32 bits
`double`	double-precision floating point	$\pm 1.7 \times 10^{+308} \ldots \pm 4.9 \times 10^{-324}$ with at least 15 decimal digits of precision	64 bits
`boolean`	Boolean	`false` or `true`	8 bits

Variable Declarations

Description	General Form	Example
Declare one variable and do not give it an initial value	*type variable*;	`int x;`
Declare one variable and give it an initial value	*type variable = value*;	`int x = 5;`
Declare multiple variables of the same type and give none of them initial values	*type variable*$_1$, *variable*$_2$, ..., *variable*$_n$;	`int x, sum, height;`
Declare multiple variables of the same type and give some or all of them initial values	*type variable*$_1$ = *value*$_1$, *variable*$_2$ = *value*$_2$, ..., *variable*$_n$ = *value*$_n$;	`int x = 5, sum, height = 0;`

Mathematical Operators

Operator	Meaning
+	addition
−	subtraction
*	multiplication
/	division
%	modulus

(All the C/C++ bitwise operators too)

Mixed Arithmetic

- Less dominant values are promoted to the dominating type to perform arithmetic

`byte short char int long float double`
⇐ **Narrower** **Wider** ⇒
⇐ **Less Dominant** **More Dominant** ⇒

Java Reserved Words

abstract	else	int	strictfp
boolean	enum	interface	super
break	extends	long	switch
byte	false	native	synchronized
case	final	new	this
catch	finally	null	throw
char	float	package	throws
class	for	private	transient
const	goto	protected	true
continue	if	public	try
default	implements	return	void
do	import	short	volatile
double	instanceof	static	while

Console Output

- **`System.out.println`**
- **`System.out.print`**
- **`System.out.printf`**

Console Input

- **`Scanner`** class
- Unlike with output, the programmer must create a **`Scanner`** object to get user input
- Use a **`System.in`** object to create the **`Scanner`** object
- Can use a file reader object to create a **`Scanner`** object to read from a text file instead of the keyboard

Graphical I/O

- **`JOptionPane`**
  - **`showMessageDialog`**
  - **`showInputDialog`**