

Lecture Set #3: Java Expressions

- Last time:
1. Basics of Java programs
- Today:
1. Variables and types
 2. Expressions in Java
 3. User input



CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

Variables ...

- ... are named storage locations

Variable	Value
x	5

- Recall that memory is a sequence of bits
- Question: How much memory to allocate for a variable's value?
- Answer: A variable must have a *type* specifying how much storage to allocate.

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

1



Recall Java Built-in Types

	Type name	Size (bytes)
Integers	byte	1
	short	2
	int	4
	long	8
Reals	float	4
	double	8
Other	char	2
	boolean	1

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

2



Primitive Data Types In Detail



Integer Types:

byte	1 byte	Range: -128 to +127
short	2 bytes	Range: -32,000 to +32,000
int	4 bytes	Range: -2 billion to +2 billion
long	8 bytes	Range: -9 quintillion to +9 quintillion

Floating-Point Types:

float	4 bytes	-3.4x10 ³⁸ to 3.4x10 ³⁸ , 7 digits of precision
double	8 bytes	-1.7x10 ³⁰⁸ to 1.7x10 ³⁰⁸ , 15 digits of prec.

Other types:

boolean	1 byte	true, false
char	2 bytes	A single (Unicode) character

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

3

Primitive-Type Constants



- Constants are also called *literals*

Integer types:

byte	} optional sign and digits (0-9): 12 -1 +234 0 1234567
short	
int	
long	

Same as above, but followed by 'L' or 'l': -1394382953L

Floating-point types:

double	Two allowable forms:	Avoid this lowercase L. It looks too much like the digit '1'
Decimal notation:	3.14159 -234.421 0.0042 -43.0	
Scientific notation:	(use E or e for base 10 exponent)	
	3.145E5 = 3.145 x 10 ⁵ = 314500.0	
	1834.23e-6 = 1834.23 x 10 ⁻⁶ = 0.00183423	
float	Same as double, but followed by 'f' or 'F': 3.14159F -43.2f	

Note: By default, integer constants are **int**, unless 'L'/'l' is used to indicate they are **long**. Floating constants are **double**, unless 'F'/'f' is used to indicate they are **float**.

Jan Plane (adapted from Bonnie Dorr)

4

Character and String Constants



- Char constants:** Single character in single quotes ('...') including:

- Letters and digits: 'A', 'B', 'C', ..., 'a', 'b', 'c', ..., '0', '1', ..., '9'
- Punctuation symbols: '!', '#', '@', '\$' (except ' and backslash '\')
- Escape sequences: (see below)

- String constants:** 0 or more characters in double quotes ("...")

- Escape sequences:** Allows inclusion of special characters:
 - '\ ' double quote '\n' new-line character (start a new line)
 - '\ ' single quote '\t' tab character
 - '\ ' backslash

- Examples:** `char x = '\ ';` → (x contains a single quote)
`String s1 = "\Hi there!\\";` → s1 contains "Hi there!"
`String s2 = "C:\\WINDOWS";` → s2 contains C:\WINDOWS

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

5

User Input in Java



- We've done output (System.out); what about input?
- Java 5.0 includes the **Scanner class** feature
 - Can use Scanner to create "scanner objects"
 - Scanner objects convert user input into data
- To use Scanner need to *import* a library:
`import java.util.Scanner;`

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

9

Example5.java



```
import java.util.Scanner;

public class Example5 {

    public static void main(String[] args) {
        int i;
        double d;
        String s;

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter an integer: ");

        i = sc.nextInt();
        System.out.print("Enter a floating point value: ");

        d = sc.nextDouble();
        System.out.print("Enter a string: ");

        s = sc.next();

        System.out.println("Here is what you entered: ");
        System.out.println(i);
        System.out.println(d);
        System.out.println(s);
    }
}
```

Include the definition of the Scanner utility

Create new scanner object to read from keyboard

Input an integer

Input a double

Input a string (up to white space)

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

10

Scanner Class Details



- To create a scanner object:
`new Scanner(input_source);`
 - Input source can be keyboard (System.in), files, etc.
 - Object must be assigned to a variable (e.g. sc)
- Operations
 - nextBoolean()
 - nextByte()
 - nextDouble()
 - nextFloat()
 - nextInt()
 - nextLong()
 - nextShort()

Returns value of indicated type (reports error if type mismatch)

 - next() Returns sequence of characters up to next whitespace (space, carriage return, tab, etc.)
 - nextLine() Returns sequence of characters up to next carriage return

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

11

Debugging Java Programs



- Types of errors
 - "Compile time": caught by Eclipse / Java compiler
 - *Syntax* errors: typos, etc.
 - *Type* errors: misuse of variables
 - "Run time": appear during program execution
 - Division by 0
 - Wrong outputs (because of mistakes in programming)
- Eclipse helps catch compile time errors
 - **Red**: error
 - **Yellow**: warning

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

12

Example3.java – find the errors



```
public class Example3 {  
  
    public static void main(String[] args) {  
        int x = 7;  
        int y = 12;  
        double d = 72.33;  
        boolean b = true;  
        char c;  
        String s;  
  
        x = y + 24;  
        y = 17.3;  
        d = x;  
        b = 17;  
        c = "cow";  
        s = "Here is something weird " + x + y;  
    }  
}
```

CMSC 131 Spring 2007
Jan Plane (adapted from Bonnie Dorr)

13
