

Chapter 2: Beginning to Program

CS1: Java Programming
Colorado State University

Original slides by Daniel Liang
Modified by Kris Brown, Ben Say, Wim Bohm



Announcements

Upcoming ACM & ACM-W Events

Internship Panel 29 Jan
CSB 130 @ 6pm

HPE Data Center Visit 7 Feb
<https://forms.gle/pAfp1FRS3eQXLfWs7>

T9Hacks 7 - 8 Feb
<https://forms.gle/xkpTkReR59TZgsZE6>

RamRide Fundraiser 14 Feb
<https://forms.gle/vf2F27K1k6DZovWV9>

HackCU 22 Feb
<https://forms.gle/fv1gkSwqboTP6DTp6>

Motivations

- Solve practical problems programmatically
- Java primitive data types
- Strings
- Input/Output
- Constants



Variables

A named container that holds a specific piece of data.

Variables have a type (set of values). Some Java types are:

int, double, char, String
(more later)



Declaring Variables

```
int x;           // Declare x to be an
                 // integer variable;

double radius;  // Declare radius to
                 // be a double variable;

char a;         // Declare a to be a
                 // character variable;

String s;       // Declare s to be a
                 // String variable;
```



Assignment Statements

```
x = 1;           // Assign 1 to x;  
radius = 1.0;   // Assign 1.0 to radius;  
a = 'A';        // Assign 'A' to a;  
s = "Java";     // Assign "Java" to s
```



Declaring and Initializing in One Step

- `int x = 1;`
- `double d = 1.4;`
- `String s = "Java";`



Variable names

- A variable name is a sequence of characters that consist of letters, digits, underscores (`_`), and dollar signs (`$`).
- A variable name must start with a letter, an underscore (`_`), or a dollar sign (`$`). It cannot start with a digit.
- A variable name cannot be a reserved word. (See Appendix A, “Java Keywords,” for a list of reserved words).
- A variable name cannot be `true`, `false`, or `null`.
- A variable name can be of any length.



Numerical Data Types

Name	Range	Storage Size
byte	-2^7 to $2^7 - 1$ (-128 to 127)	8-bit signed
short	-2^{15} to $2^{15} - 1$ (-32768 to 32767)	16-bit signed
int	-2^{31} to $2^{31} - 1$ (-2147483648 to 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

Printing

```
System.out.println("Hello World");
```

- get the computer to print something to the console
- println prints a line and adds a new line at the end
- print prints the line and continues on the same line
- use for **DEBUGGING!!**



Simple String Operations

Concatenation:

Use the “+” (plus sign) to concatenate strings

```
System.out.println(mm + " " + yy);
```



Simple String Operations

The *length()* method

```
String theName = "Donald Duck";  
int len = theName.length();
```

What is returned by `length()` ?



Reading Input from the Console

1. Create a Scanner object

```
Scanner input = new Scanner(System.in);
```

2. Use the method `nextDouble()` to obtain to a double value. For example,

```
System.out.print("Enter a double value: ");  
Scanner input = new Scanner(System.in);  
double d = input.nextDouble();
```

Let's play with IO in Eclipse



Reading Numbers from the Keyboard

```
Scanner input = new Scanner(System.in);  
int value = input.nextInt();
```

Method	Description
<code>nextByte()</code>	reads an integer of the <code>byte</code> type.
<code>nextShort()</code>	reads an integer of the <code>short</code> type.
<code>nextInt()</code>	reads an integer of the <code>int</code> type.
<code>nextLong()</code>	reads an integer of the <code>long</code> type.
<code>nextFloat()</code>	reads a number of the <code>float</code> type.
<code>nextDouble()</code>	reads a number of the <code>double</code> type.



Variables

```
// Compute the first area
```

```
radius = 1.0;
```

```
area = radius * radius * 3.14159;
```

```
System.out.println("The area is " +  
    area + " for radius "+radius);
```

```
// Compute the second area
```

```
radius = 2.0;
```

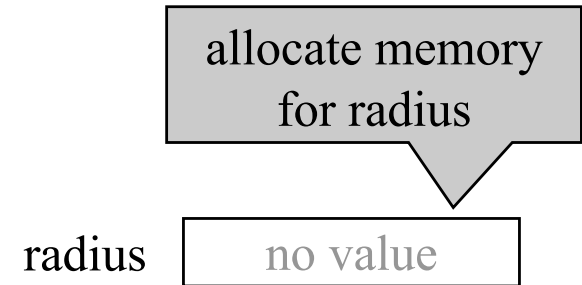
```
area = radius * radius * 3.14159;
```

```
System.out.println("The area is " +  
    area + " for radius "+radius);
```



Trace a Program Execution

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```



Trace a Program Execution

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

memory

radius

area

allocate memory
for area



Trace a Program Execution

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

radius

assign 20 to radius

20.0

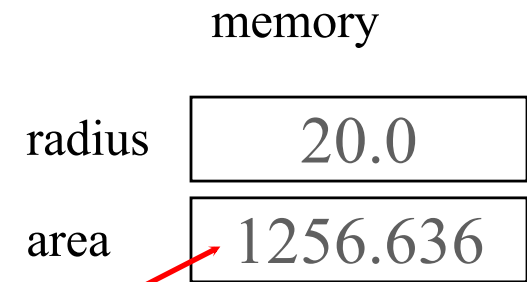
area

no value



Trace a Program Execution

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```



compute area and assign it to variable area



Trace a Program Execution

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

memory

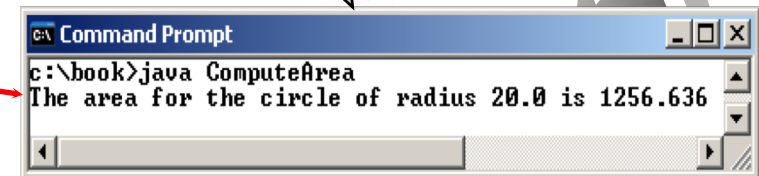
radius

20.0

area

1256.636

print a message to the console



Your Turn!

Write code in which a *String* variable *message* contains “The number of rabbits is”. An integer variable *num* has a value of 129. Concatenate these variables into a *String* called *report*. Then print, using *report*:

```
The number of rabbits is 129!!
```

