CSE 20
Intro to Computing I
Lecture 2 – Variables

Announcements

- Cyber Aware Day
  - Friday (10/13) 10am – 3:15pm California Room
  - http://cyberaware.ucmerced.edu/
- Lab #3 this week
  - Due in a week
  - Make sure to show your work to **YOUR TA** (or me) before submission
- Reading assignment
  - Chapter 2.6 – 2.10 of textbook

PEER ASSISTED LEARNING SUPPORT

- Go to **learning.ucmerced.edu**
- Click on **“Programs”**
- Scroll down and click on **Peer Assisted Learning Support (PALS) to find out more**
- Click on the **“Learning Support Schedule”**

OR

use this shortcut to go straight to the schedule:

http://bit.ly/PALS_Schedule

“Peer Assisted Learning Support,
Your learning community.”

Outputs in Java

- System.out.print(“Test print”);

Output:

```
Test print     (Doesn’t end with a newline)
```

Outputs in Java

- System.out.print(“Test print”);
- System.out.println(“Test println”);

Output:

```
Test printTest println   (Ends with a newline)
```

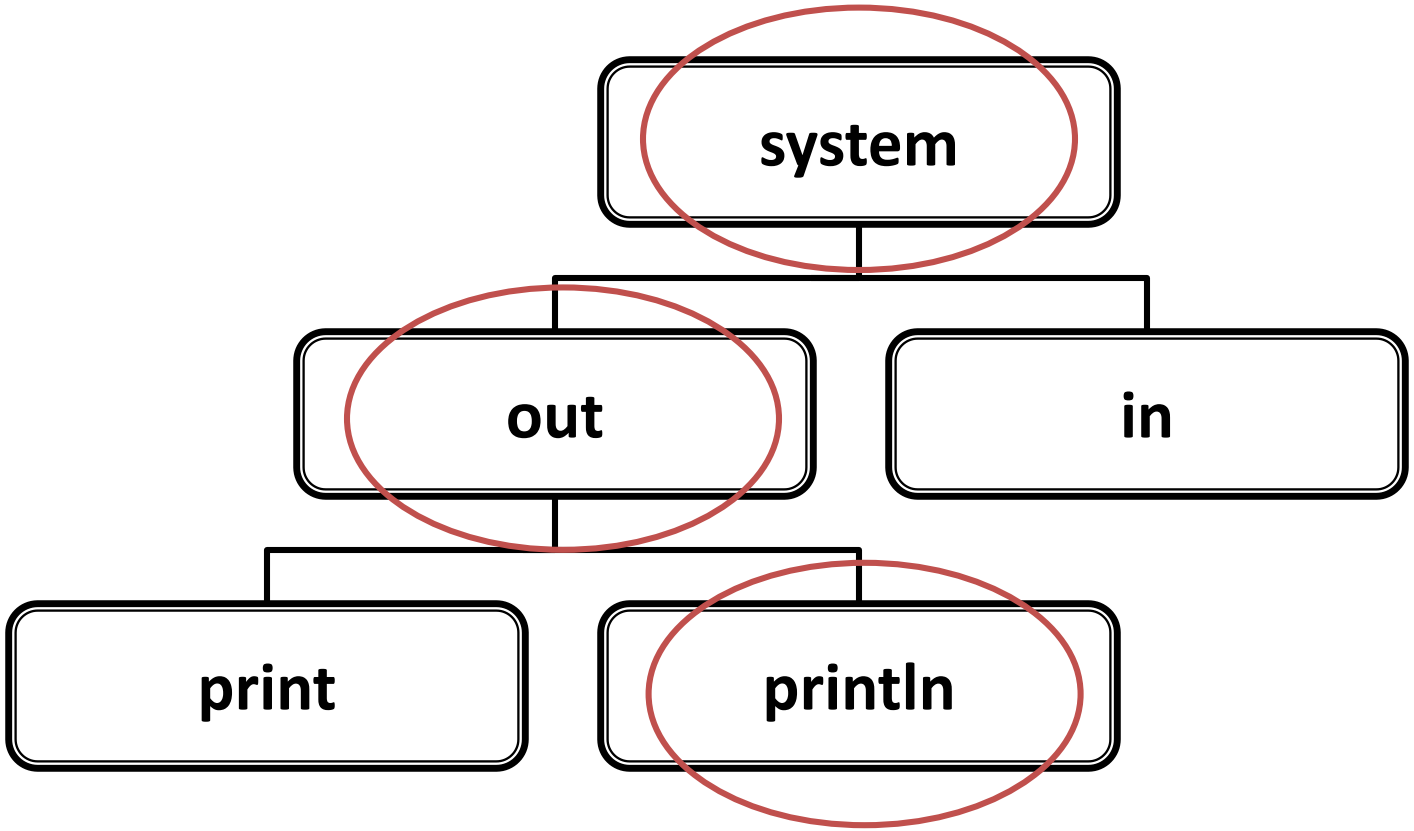
Outputs in Java

- System.out.print(“Test print”);
- System.out.println(“Test println”);
- System.out.print(“Done”);

Output:

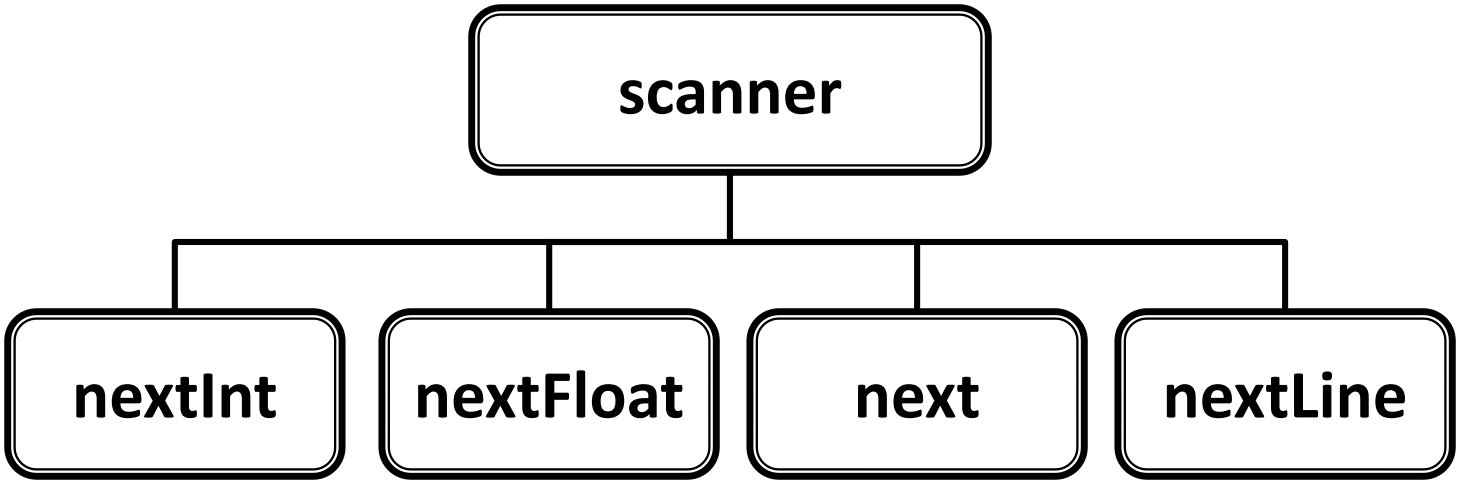
```
Test printTest println
Done
```

System Objects




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

Input – Scanner




```
Scanner input = new Scanner(System.in);
input.nextInt();
input.nextFloat();
input.next();
input.nextLine();
```

How well do you know …

- What type of files do you have in your computer?
  - Essays in Word Documents?
    - 25-100 KB (Kilo-Bytes)
  - Music - MP3's?
    - 3-5 MB (Mega-Bytes)
  - Movies – mov, mp4?
    - 2-4 GB (Giga-Bytes)
- How fast is your internet connection?
  - DSL
    - 356 Kbps to 6 Mbps (Kilo-/Mega-bits per second)
  - Cable
    - ~ 6-15 Mbps
- How long does it take to download 5 MB file using DSL of 1 Mbps?

Need to convert Bytes into bits!

What are they?

What's in a bit?

- Could be used to represent two values :
  - 0 or 1 (Off or On)
- How many values would 2 bits take on?
  - 00
  - 01
  - 10
  - 11
- In computer, information is always stored as power of 2's
  - N bits → 2^N possible values
- Byte is the basic unit in computer storage
  - 1 Byte = 8 bits

Data Types

- **Boolean** : 1-bit
  - 2 values, range : 0-1
- **Short** : 16-bits (2 bytes)
  - 2^{16} values, range : -32,768 to 32,767
- **Char**: 16-bits (2 bytes)
  - 2^{16} values, range : 0 to 65,535
- **Int** : 32-bits (4 bytes)
  - 2^{32} values, range : -2,147,483,648 to +2,147,483,647
- **Float**: 32-bits (4 bytes)
  - Scientific format : $\pm 3.4 \times 10^{\pm 38}$
- **Double** : 64–bits (8 bytes)
  - $\pm 1.7 \times 10^{\pm 308}$
- **String** : Any length (string of characters)

Numbers : Operations

Arithmetic operator	Description
+	addition
-	subtraction
*	multiplication
/	division
%	modulo (remainder)

Using Numbers

Using Numbers

- Add names/identifiers to each as a way of referring to them.

Using Numbers

- Add names/identifiers to each as a way of referring to them.
  - They can be any word.
  - Try to choose the names that **make sense**.

Variables

- Add names/identifiers to each as a way of referring to them.
  - They can be any word.
  - Try to choose the names that make sense.
- Need to know the data types.

Variables

- “first” can be of what type?
  - Short, integer, float, double
- “second” can be of what type?
  - Float, double
- “result” can be of what type?
  - Float, double

Code – Executable Statements

```
int first = 1;
double second = 0.5;
double result = first - second;
```

Code – Executable Statements

```
int first; // Declaration (type name)
first = 0; // Assignment (initialize)
first = 1; // Assignment (reuse/override)
double second = 0.5; // Declare + Assign
double result = first - second;
```

Type Casting (Up Conversion)

```
double first; // use "higher" Type
first = 0; // 0 is also a valid double (0.0)
first = 1; // 1.0
double second = 0.5;
double result = first - second;
```

Type Casting (Down Conversion)

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
// using "lower" Type needs explicit cast
```

Output Variable

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
System.out.println(result);
```

Console Output: 0

Output Message

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
System.out.println("Result is ");
System.out.println(result);
```

Console Output: Result is
0

Output Message – Corrected

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
System.out.print("Result is ");
System.out.println(result);
```

Console Output: Result is 0

Output Message using +

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
System.out.println("Result is " + "result");
```

A string!!!

Console Output: Result is result

Output Variable using +

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
System.out.println("Result is " + result);
```

Console Output: Result is 0

String Variable (1)

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
String outMessage = "Result is ";
System.out.println(outMessage + result);
```

Console Output: Result is 0

String Variable (2)

```
double first;
first = 0;
first = 1;
double second = 0.5;
int result = (int)(first - second);
String outMessage = "Result is " + result;
System.out.println(outMessage);
```

Save output string as outMessage first.
Then print it out.

Console Output: Result is 0

Type Conversions

- Implicit – Up (no need to specify, no information loss)
  - double d = 4;
  - int i = 'A';
  - float f = 'A';
  - double e = 'A';
- Explicit – Down (need to specify, may introduce information loss)
  - a = (char)i;
  - a = (char)f;
  - a = (char)d;
  - i = (int)f;
  - i = (int)e;
  - f = (float)e;

Addition : + (Data Types)

- short + short → int
- short + int → int
- char + char → int
- int + int → int
- int + float → float
- string + boolean → string
- string + (expression) → string
- string + char + char → string + char → string
- char + char + string → int + string → string

Highest data type in the expression

Names are Case Sensitive

- MAIN
- Main
- main
- mAin
- maIn
- maiN
- mAIn
- MaiN
- Everything above is a different “word”!

Naming Convention

- Begin with letter or _
- Class names capitalized
  - Averages
  - FirstProgram
- Variable names
  - Begins with lowercase letter
    - main
    - average
    - result
  - Combining words
    - toUpper
    - toUpperCase
    - theSquare

Putting it all Together

```
Scanner input = new Scanner(System.in); //Allow user input
System.out.print("What is your name? ");
String name = input.next();
System.out.print("Where do you live " + name + "? ");
String city = input.next();

System.out.println("\n" + name + " lives in " + city + ".");
```

```
Output:
What is your name? Daniel
Where do you live Daniel? Merced
Daniel lives in Merced.
```