

INPUT VALIDATION

CS302 – Introduction to Programming
University of Wisconsin – Madison
Lecture 8

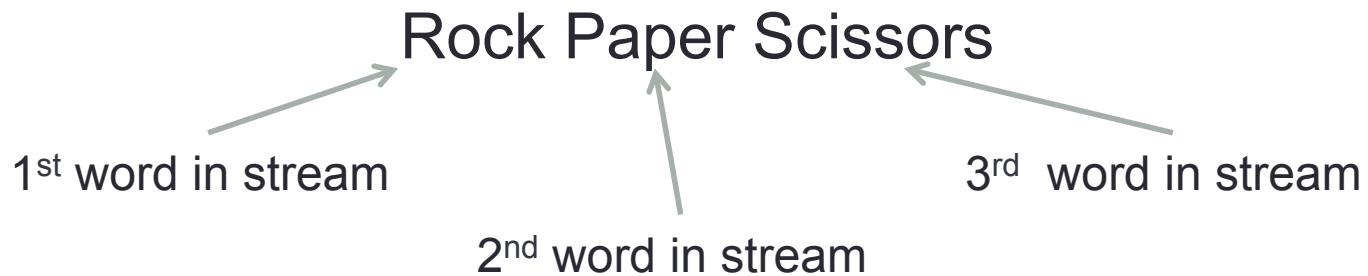
By Matthew Bernstein – matthewb@cs.wisc.edu

Input Validation – Type Checking

- Scanner provides us with methods that allow us to check that the type of data being input by the user is what we are expecting to receive in our program
- Useful Methods:
 - `Scanner.hasNextInt()`
 - `Scanner.hasNextDouble()`
 - `Scanner.nextLine()`

How Scanner Works

- We think of the Scanner as a machine we use for processing a stream of words
- The user input is treated as a stream where by default, words are separated by spaces (although you can change this if you wanted to)
- Example:



next()

- This method just takes the next word in the stream, and returns it as a String
- If we have the following code:

```
String word = scan.next();
```

- If the user input the following:

```
rock paper scissors
```

- Our variable word would store “rock” because it is the first word in the stream

next()

- If we have the following code:

```
String word = scan.next()  
word = scan.next()
```

- And the user inputs the following:

rock paper scissors

- Our word variable would now store “paper” because we have already processed the first word in the stream (“rock”) with the first call to `next()` and we now move to the next word (“paper”) with the second call to `next()`

Blocking as we wait for input

- If we try to use `next()` (or any of its variants), but there are no words in the current stream, then this method will block execution of the program and wait until it gets at least one word in its stream that it can process.
- We are very familiar with this concept by now

Using the “hasNext” methods

- If we want to validate the type of input entered by the user we would use one of the “hasNext” methods
- `hasNextInt()` → this method returns a boolean value (true or false) that tells us whether the next word in its input stream is a valid integer
- `hasNextDouble()` → this method returns a boolean value that us whether the next word in its stream is a valid integer
- **HOWEVER**, these methods do not advance the Scanner past the next word in the stream. It simply gives us some information about that word.
- These methods **WILL** block execution if stream is empty

Back to Loops – Processing Sentinel Values

- Let's say we have a scenario in which we are reading a sequence of inputs. How do we indicate when the sequence ends?
- We use a **sentinel value** to signal the end of the sequence.
- In our Rock, Paper, Scissors example we used the string “q” to signal that the sequence of hands we are throwing is finished

Another Example of Sentinel Values

```
while (salary != -1)
{
    Salary = in.nextDouble();
    if (salary != -1)
    {
        sum = sum + salary;
        count++;
    }

    if (count > 0)
    {
        double average = sum / count;
        System.out.println("Average salary: " + average);
    }
    else
    {
        System.out.println("No data");
    }
}
```

Cool CS Link of the Day

- TED Talk by Blaise Aguera y Arcas: How PhotoSynth can connect the world's images
- http://www.ted.com/talks/blaise_aguera_y_arcas_demos_photosynth.html

