The Scanner Methods

#1 .nextInt( ) Searches user entry for the next integer that is entered, returns integer (AND STOPS ONCE FOUND) …

User Enters	**Computer Grabs**
67	67 … but does not grab the "enter" newline that user typed, stops at integer!
67	67 … computer ignores all spaces and newlines and only grabs the integer!!!
I am 67	ERROR … nextInt( ) can only ignore spaces/newlines and grab integers!
67.8	ERROR … nextInt( ) has no memory for decimals, InputMismatchException!
67 9.4	67 … it got the integer and looks no further (short-circuit evaluation?)!
67 I am	67 … it got the integer and looks no further (short-circuit evaluation?)!

CRUCIAL NOTE: Once the scanner grabs the integer, IT STOPS THERE. This is very important!

#2 .nextDouble( ) Searches user entry for the next double that is entered, returns double (STOPS ONCE FOUND) …

User Enters	**Computer Grabs**
3.14	3.14 … but does not grab the newline as it stops once the double is found!
3.14	3.14 … the scanner ignores all leading spaces and only grabs the double!
Pi is 3.14	ERROR … nextDouble( ) can only ignore spaces/newlines and grab doubles!
3	3.0 … remember that integers can be placed into double variables!
3.14 8	3.14 … it got the double and looks no further (short-circuit evaluation?)!
3.14 is Pi	3.14 … it got the double and looks no further (short-circuit evaluation?)!

CRUCIAL NOTE: Once the scanner grabs the double, IT STOPS THERE. This is very important!

#3 Combination of .nextInt( ) and .nextDouble: Proof of the CRUCIAL NOTES shown above:

Consider the following code:

```
Scanner scan = new Scanner(System.in);
System.out.print("Enter an integer followed by a double: ");
 int userInt = scan.nextInt();
double userDouble = scan.nextDouble();
System.out.println("You typed: "+userInt+" and "+userDouble);
```

If the user types: 67 3.14
The output will be: You typed 67 and 3.14

If the user types: 67 //they hit "enter" after typing 67
3.14
The output will be: You typed 67 and 3.14 //this occurs since the 2nd nextInt( ) call
//will pass over the newline to the double

#4 .nextLine( ) Starts grabbing everything until a newline occurs, returns String (and grabs the newline with it!)

User Enters	**Computer Grabs**	
67	67	… and the newline is also grabbed
3.14	3.14	… and the newline is also grabbed
Pi is 3.14	Pi is 3.14	… and the newline is also grabbed
Java & * 34. 1	Java & * 34.1	… and the newline is also grabbed

#5 .next( ) Grabs the very next set of characters that are typed until the next space that is entered (no newline!)

User Enters	**Computer Grabs**	
67	67	… stops there, but 67 is a String (does not grab newline)
3.14	3.14	… stops there, but 3.14 is a String (does not grab newline)
Pi is 3.14	Pi	… stops there, Pi is a String (does not grab newline)

Common Issues with Scanner Methods (Warning: .nextLine( ) can create confusion if you are not careful!) …

Example #1 Consider the following code:

```
Scanner scan = new Scanner(System.in);
System.out.print("Enter first name: ");
String userInt = scan.next();
System.out.println("Enter last name: ");
String user2  = scan.nextLine();
System.out.println("You typed: "+userInt+" and "+user2);
```

Here is the output window:

```
Enter first name: Dr.
Enter last name:
You typed: Dr. and
```

IMPORTANT NOTE: The scan.next( ) grabbed the user entry of "Dr." and then stopped. When the scan.nextLine( ) is called it picks up where the scan.next( ) left off and continues on to the newline … this means that the scan.nextLine( ) actually just picked up nothing but the newline and therefore the user never even got to enter their last name.

Example #2 Consider the following Code:

```
Scanner scan = new Scanner(System.in);
System.out.print("Enter your age: ");
int userInt = scan.nextInt();
System.out.println("Enter full name:  ");
String userName  = scan.nextLine();
System.out.println("Hello: "+userName+" who is "+userInt);
```

Here is the output window:

```
Enter your age: 67 is me
Enter full name:
You typed: 67 and  is me
```

IMPORTANT NOTE: The scan.nextInt( ) grabbed the 67 and then stopped there. There is no error thrown even though the user typed text after the 67 since the scan.nextInt( ) stops looking once it finds the 67. When the scan.nextLine( ) is called, it just picks up at the end of the 67. It finds a space and continues until the end of the line … so it finds " is me" and stores it as the last name. The user never even gets to enter their last name!

To avoid the .nextLine( ) issues, consider doing one of the following options …

1. Have the user enter all text first and then collect numbers (this avoids any # scan "stalls").
2. Use an extra .nextLine( ) between reading a number and reading text so that you grab the newline after the user's entered number (integer or double).