Simple Input

- Interactive Input - The Class `Scanner`
- Command-Line Arguments
Sometimes data is needed and obtained from the user at run time.

Simple keyboard input requires:

```java
import java.util.Scanner;
or import java.util.*;
```

at the beginning of the file.
A “Scanner” object must be initialized before inputting data:

```java
Scanner keyboard = new Scanner(System.in);
```

To input data:

```java
int eggsPerBasket;
:
eggsPerBasket = keyboard.nextInt();
```

which reads one `int` value from the keyboard and assigns it to the variable `eggsPerBasket`. 
import java.util.Scanner;

public class Cows {
    public static void main (String[] args) {
        int barns, cowsPer, totalCows;

        Scanner kb = new Scanner(System.in);
        System.out.print(“Enter the number of barns: “);
        barns = kb.nextInt();
        System.out.print(“Enter the number of cows per barn: “);
        cowsPer = kb.nextInt();

        totalCows = barns * cowsPer;
        System.out.println(“Total number of cows is: “ + totalCows);
        System.out.println(“Suppose two cows per barn escape!“);
        cowsPer = cowsPer - 2;
        totalCows = barns * cowsPer;
        System.out.println(“Total number of cows is now: “ + totalCows);
    }
}

> javac Cows.java
> java Cows
Enter the number of barns: 10
Enter the number of cows per barn: 6
Total number of cows is: 60
Suppose two cows per barn escape!
Total number of cows is now: 40
**Command-Line Arguments**

- Frequently input is provided to a program at the command-line.

```java
public class UseArgument {
    public static void main(String[] args) {
        System.out.print("Hi, ");
        System.out.print(args[0]);
        System.out.println(". How are you?");
    }
}
```

```bash
> javac UseArgument.java
> java UseArgument Alice
Hi, Alice. How are you?
> java UseArgument Bob
Hi, Bob. How are you?
```
Frequently input is provided to a program at the command-line.

public class UseArgument
{
    public static void main(String[] args)
    {
        System.out.print("Hi, ");
        System.out.print(args[0]);
        System.out.println(". How are you?");
    }
}

> javac UseArgument.java
> java UseArgument Alice
Hi, Alice. How are you?
> java UseArgument Bob
Hi, Bob. How are you?
Frequently input is provided to a program at the command-line.

```java
public class UseArgument {
    public static void main(String[] args) {
        System.out.print("Hi, ");
        System.out.print(args[0]);
        System.out.println(". How are you?");
    }
}
```

```bash
> javac UseArgument.java
> java UseArgument Alice
Hi, Alice. How are you?
> java UseArgument Bob
Hi, Bob. How are you?
```
Frequently multiple values are provided at the command-line.

```java
public class Use3Arguments {
    public static void main(String[] args) {
        System.out.print("The first word is ");
        System.out.print(args[0]);
        System.out.print(" , the second is ");
        System.out.print(args[1]);
        System.out.print(" , and the third is ");
        System.out.println(args[2]);
    }
}
```

> javac Use3Arguments.java
> java Use3Arguments dog cat cow
Frequently multiple values are provided at the command-line.

```java
public class Use3Arguments
{
    public static void main(String[] args)
    {
        System.out.print(“The first word is “);
        System.out.print(args[0]);
        System.out.print(“, the second is “);
        System.out.print(args[1]);
        System.out.print(“, and the third is “);
        System.out.println(args[2]);
    }
}
```

> javac Use3Arguments.java
> java Use3Arguments dog cat cow
Frequently multiple values are provided at the command-line.

```java
public class Use3Arguments {
    public static void main(String[] args) {
        System.out.print("The first word is ");
        System.out.print(args[0]);
        System.out.print(", the second is ");
        System.out.print(args[1]);
        System.out.print(", and the third is ");
        System.out.println(args[2]);
    }
}
```

> javac Use3Arguments.java
> java Use3Arguments dog cat cow

The first word is dog, the second is cat, and the third is cow
What if you provide integers on the command-line?

```java
public class IntOps {
    public static void main(String[] args) {
        int sum, prod, quot, rem;
        sum = args[0] + args[1];
        prod = args[0] * args[1];
        quot = args[0] / args[1];
        rem = args[0] % args[1];
        System.out.println("Sum = " + sum);
        System.out.println("Product = " + prod);
        System.out.println("Quotient = " + quot);
        System.out.println("Remainder = " + rem);
    }
}
```

Hoping to see this

```
> javac IntOps.java
> java IntOps 1234 99
Sum = 1333
Prod = 122166
Quotient = 12
Remainder = 46
```
What if you provide integers on the command-line?

```java
public class IntOps {
    public static void main(String[] args) {
        int sum, prod, quot, rem;
        sum = args[0] + args[1];
        prod = args[0] * args[1];
        quot = args[0] / args[1];
        rem = args[0] % args[1];
        System.out.println("Sum = "+sum);
        System.out.println("Product = "+prod);
        System.out.println("Quotient = "+quot);
        System.out.println("Remainder = "+rem);
    }
}
```

```
javac IntOps.java
*** ERROR ***
```
For the same reason, this does NOT work:

```java
public class IntOps {
    public static void main(String[] args) {
        int a, b, sum, prod, quot, rem;
        a = args[0];
        b = args[1];
        sum = a + b;
        prod = a * b;
        quot = a / b;
        rem = a % b;
        System.out.println("Sum = " + sum);
        System.out.println("Product = " + prod);
        System.out.println("Quotient = " + quot);
        System.out.println("Remainder = " + rem);
    }
}
```

```bash
> javac IntOps.java
*** ERROR ***
```
So, it looks ugly, but this is what we need to do:

```java
public class IntOps {
    public static void main(String[] args) {
        int a, b, sum, prod, quot, rem;
        a = Integer.parseInt(args[0]); // Notice the Integer.parseInt
        b = Integer.parseInt(args[1]); // Notice the comments...lol
        sum = a + b;
        prod = a * b;
        quot = a / b;
        rem = a % b;
        System.out.println("Sum = " + sum);
        System.out.println("Product = " + prod);
        System.out.println("Quotient = " + quot);
        System.out.println("Remainder = " + rem);
    }
}
```

```
> javac IntOps.java
> java IntOps 1234 99
Sum = 1333
Prod = 122166
Quotient = 12
Remainder = 46
```