CMPS 200: Introduction to Programming Using JAVA

LECTURE 3 – Interactive Programs

Maurice J. KHABBAZ, Ph.D.

Last Time

- Programming skills development.
- Preliminary computer system organization and architecture.
- Computations, types of knowledge, algorithm development.
- Aspects of programming languages, language levels, writing code.
- JAVA basics
- Mathematical operations.
- JAVA variables and types.

Today

Interactive Programs: The `Scanner` Class

Interactive Programs

- Programs generally require inputs on which to operate.
- Often, programs designed to read data interactively during execution.
- This way:
  - New results are computed each time the program is executed.
  - Results will depend on the data that is entered.
  - Entered results will be stored in variables declared throughout the program.
- **!! Attention !!** Users can be unpredictable and may misbehave.

- **Q:** How to design programs to read data interactively from the user?

The **Scanner** Class

Part of the standard JAVA class library.

- Included in the `java.util` package (need to import this package to use the `Scanner` class).
- **Syntax import a package:** `import <package_name>;`
- **Example:** `import java.util.Scanner;`
- More on importing packages and modules later.

Provides convenience methods for reading input values of various types.

Input may come from various sources:

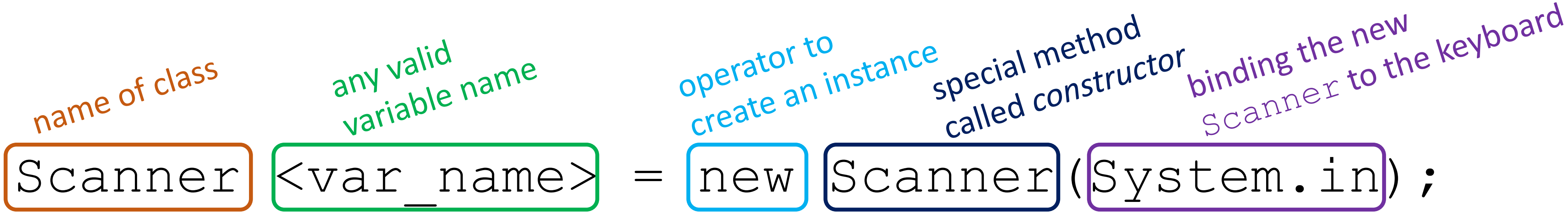
- **Now:** focus on data typed interactively by user.
- **Later:** learn how to read data stored in a file.

Allows to parse character strings into separate pieces:

- Part of the so-called string processing.

`Scanner` Object

- Can read inputs from various sources:
  - Particularly, can be setup to **communicate with/read input from the keyboard:**
    - **Keyboard input** is a.k.a **console input.**
  - Keyboard is represented in JAVA by the object **`System.in`**. standard input stream
- A `Scanner` object must first be created in order to use its methods.
- **Syntax to create a new instance of a `Scanner` object:**



```
Scanner <var_name> = new Scanner(System.in);
```

name of class

any valid variable name

operator to create an instance

special method called constructor

binding the new `Scanner` to the keyboard

- **Example:** `Scanner keyboard = new Scanner(System.in);`

`Scanner` Object

- Elements of a data input stream are called **tokens**.
- Characters that separate tokens from each other are called **delimiters**:
  - Single white space.
  - Tab (*i.e.* five white spaces).
  - New lines
- **By default** a `Scanner` object assumed that these delimiters are used:
  - The default delimiters **can be changed** using built-in methods in `Scanner` class.

- **Example:** How many tokens appear on the following line? **9**

```
23  John Smith   42.0  "Hello world"  $2.50  "  19"
```

Some Methods From the Scanner Class

Method	Return Type	Description
next()	String	Returns the next input token as a String
nextLine()	String	Returns all input remaining on current line as a String

nextBoolean()
nextByte()
nextDouble()
nextFloat()
nextInt()
nextLong()
nextShort()
	boolean
	byte
	double
	float
	int
	long
	short
		- Returns the next input token as the indicated type.
		- Throws an InputMismatchException if the next token is inconsistent with the type.

hasNext()	boolean	Returns true if the Scanner has another token in its input
useDelimiter(String pattern)	Scanner	Sets the Scanner's delimiting pattern
delimiter()	Pattern	Returns the currently used delimiting pattern

- **Remarks:**
  1. Each method waits until the user presses the ENTER key on the keyboard.
  2. The value typed by the user is returned to the method's calling point (typically an assignment)

Example 1

```java
import java.util.Scanner;

public class FirstInteractiveProgram {
    public static void main (String[] args) {
        Scanner console = new Scanner(System.in);

        System.out.print("How old are you? ");
        int age = console.nextInt();
        System.out.println("You are " + age + "year(s) old.");
    }
}
```

prompt

The **prompt** is a message indicating to the user what input to feed in.

Example 2

Write a JAVA application that takes a user’s `age` as input and outputs the number of remaining years left until retirement. Assume that an individual retires at the age of 65.

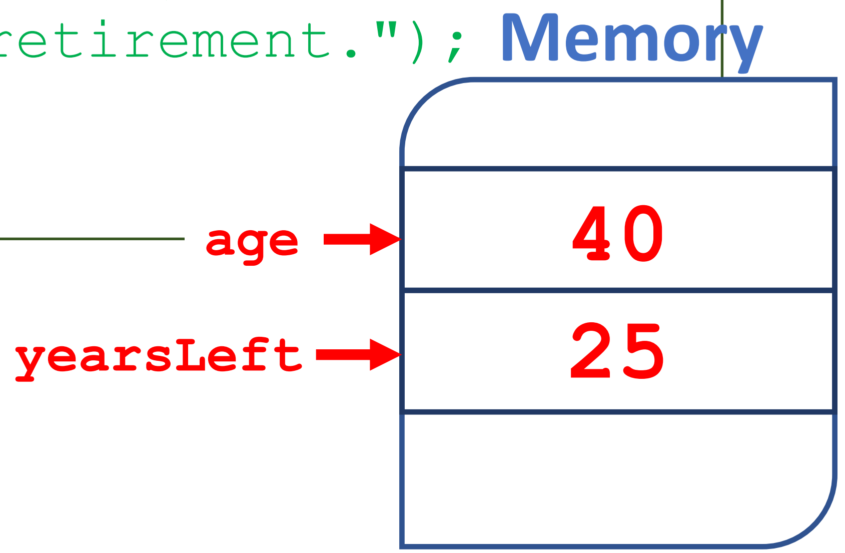
```
import java.util.Scanner;

public class FirstInteractiveProgram {
   public static void main (String[] args) {
         Scanner keyboard = new Scanner(System.in);
         int age, yearsLeft;
         System.out.print("How old are you? ");
         age = keyboard.nextInt();
         yearsLeft = 65 - age;
         System.out.print("You have " + yearsLeft + "year(s)");
         System.out.println(" until retirement.");
   }
}
```

Example 2: Sample Execution

```java
import java.util.Scanner;

public class FirstInteractiveProgram {
   public static void main (String[] args) {
      Scanner keyboard = new Scanner(System.in);
      int age, yearsLeft;
      System.out.print("How old are you? ");
      age = keyboard.nextInt();
      yearsLeft = 65 – age;
      System.out.print("You have " + yearsLeft);
      System.out.println(” year(s) until retirement.");
   }
}
```



Output to the Screen:

```
How old are you? 40⏎
You have 25 year(s) until retirement.
```

Example 3

Write a JAVA application that takes as input from the user two integers n1 and n2 entered simultaneously on the same line and outputs the result of their multiplication.

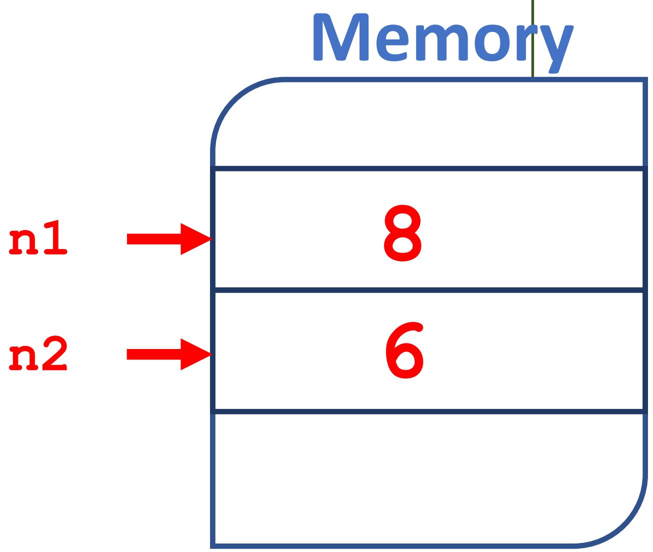
```
import java.util.Scanner;

public class Multiplication {
   public static void main (String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print(”Enter two integers: ");
        int n1 = keyboard.nextInt(), n2 = keyboard.nextInt()
        System.out.print("n1 * n2 = " + n1 * n2);
   }
}
```

Example 3: Sample Execution

```
import java.util.Scanner;

public class Multiplication {
   public static void main (String[] args) {
      Scanner keyboard = new Scanner(System.in);
      System.out.print(”Enter two integers: ");
      int n1 = keyboard.nextInt(), n2 = keyboard.nextInt();
      System.out.print("n1 * n2 = " + n1 * n2);
   }
}
```



Output to the Screen:

```
Enter two integers: 8 6⏎
n1 * n2 = 48
```

Java™