Lesson 10: Quiz #1 and Getting User Input (W03D2)

Balboa High School

Michael Ferraro

September 1, 2015

Do Now: Prep GitHub Repo for PS #1

- You'll need to submit the §5.2 solution on the paper form *and* to GitHub.
- In Eclipse, create a project called `ps01` (lowercase!).
- Add a file to the `src` folder called `Ch2Num11.java`.
- Add a class declaration and `main()` to the file.
- Use `github-push.sh` to push this project to GitHub:

  ```
  cd ~/MOUNTED/apcs-locker/workspace0/ps01
  github-push.sh GITHUB_USERNAME PERIOD
  ```

- When you work from home, change the file using your web browser.

Aim

Students will take Quiz #1 and learn how to write a Java program to prompt the user for input values via a `Scanner`.

After the Quiz

Once you've finished the quiz:

- bring your quiz to me
- remain quiet as a courtesy to those still working
- open up this set of lesson slides
- move through the remaining slides on your own
- HW is on the last slide!

Prompting the User for Data

Let's play with an interactive app! We will ask the user for two integers and then display the *product*.

Prompting the User for Data

1. Within `workspace0`, create a new Eclipse project called `InteractiveInts`
2. Download `InteractiveInts.java` from here, saving the file to your Desktop
3. Drag-and-drop the file from the Desktop to the `src` folder inside your new Eclipse project. **This is how you import downloaded files into Eclipse projects — just like you'll need to do when downloading Java source files from the problem sets or after saving email attachments.**
4. Run the program.

Inside InteractiveInts

```
import java.util.Scanner;

public class InteractiveInts {
     ... //3 dots here just means some code has been omitted f
}
```

`import` statements at the top of a Java class tell the compile where some "tool" is located. In this case, we're using a Java-provided tool called a `Scanner`, which is useful for capturing what a user types (among other things that we'll go into later).

Inside InteractiveInts

```
import java.util.Scanner;

public class InteractiveInts {

    public static void main(String[] args) {

        Scanner keybd = new Scanner(System.in);
        ...
        int firstInt = new Integer(keybd.nextLine());
        ...
    }
}
```

---

keybd is the name of a Scanner "object" we have created.

Inside InteractiveInts

```
import java.util.Scanner;

public class InteractiveInts {

    public static void main(String[] args) {

        Scanner keybd = new Scanner(System.in);
        ...
        int firstInt = new Integer(keybd.nextLine());
        ...
    }
}
```

Where you see `keybd.nextLine()`, we're asking the `Scanner` called `keybd` to read in the characters that the user types until [ENTER] is pressed.

Inside InteractiveInts

```
import java.util.Scanner;

public class InteractiveInts {

    public static void main(String[] args) {

        Scanner keybd = new Scanner(System.in);
        ...
        int firstInt = new Integer(keybd.nextLine());
        ...
    }
}
```

That typed value is turned into an integer, if possible, and then stored in the `int` variable called `firstInt`.

Things to Try

- What happens if you type `abc` and press [ENTER] when asked to provide an integer?
- How does the program behave if you provide a non-integer value at one or both prompts? For example, try entering 2.1 during one program run and 2.9 the next time.

PS #1 Tip: *Tic-Tac-Toe* Problem

Ch. 1, #16: Your job is to represent a game of tic-tac-toe using a string of 0s and 1s.

- decide how to represent `X`, `O`, and blanks[1] using bits
- include an example board and how you'd represent the game state using bits
- don't use more bits than you need
- make sure you explain your system for representing the game state in a way that your teacher will easily understand — that will affect your grade!

[1] Students commonly forget to represent blanks on the board. Imagine you're saving a game that's halfway played to continue playing later; blank locations need to be remembered.

HW

- PS #1, §5.1
  - make sure you have your Java source file available at school so you can get a sign-off
  - be prepared to explain how your program works
- Finish the remaining sections of PS #1 – **web questions due at the start of 5th Period!**
- Paper form should be turned in during next class, but will be accepted for full credit by 4:00PM on Friday. Don't procrastinate on sign-offs: Your teacher won't accept a flood of sign-offs on the last day.
- Having trouble? Post questions on the course website under the PS #1 post.