CSSE 220 Day 2

Types, Loops, and Strings

Check out *LoopsAndStrings* from SVN

Questions?

Your questions about …

- The syllabus
- The Schedule
- Java
- etc.

- Could everyone *checkout* and *commit* the HW1 project?

More announcements

- Cell Phones
  - please set ringers to silent or quiet.
    - Minimize class disruptions.
    - But sometimes there are emergencies.
- Personal needs
  - If you need to leave class for a drink of water, a trip to the bathroom, or anything like that, you need not ask me. Just try to minimize disruptions.
- Please be here and have your computer up and running by the beginning of class time as best you can.

Bonus points for reporting bugs

- In the textbook
- In any of our materials.
- More details in the Syllabus

- Check out Piazza

Some major emphases of 220

- *Reinforce from 120:*
  - Procedural programming (functions, conditionals, loops, etc)
  - Using objects
- *Object-Oriented Design*
  - Major emphasis on interfaces
  - GUI programming using Java Swing
  - UML class diagrams
- *Software Engineering concepts*
- *Recursion*
- *Program Efficiency Analysis and big-O notation*
- *Simple sorting and searching algorithms*
  - as examples for the above
- *Data Structures*
  - Abstract data types
  - Specifying and using some standard data structures
  - Implementing simple data structures (lists)

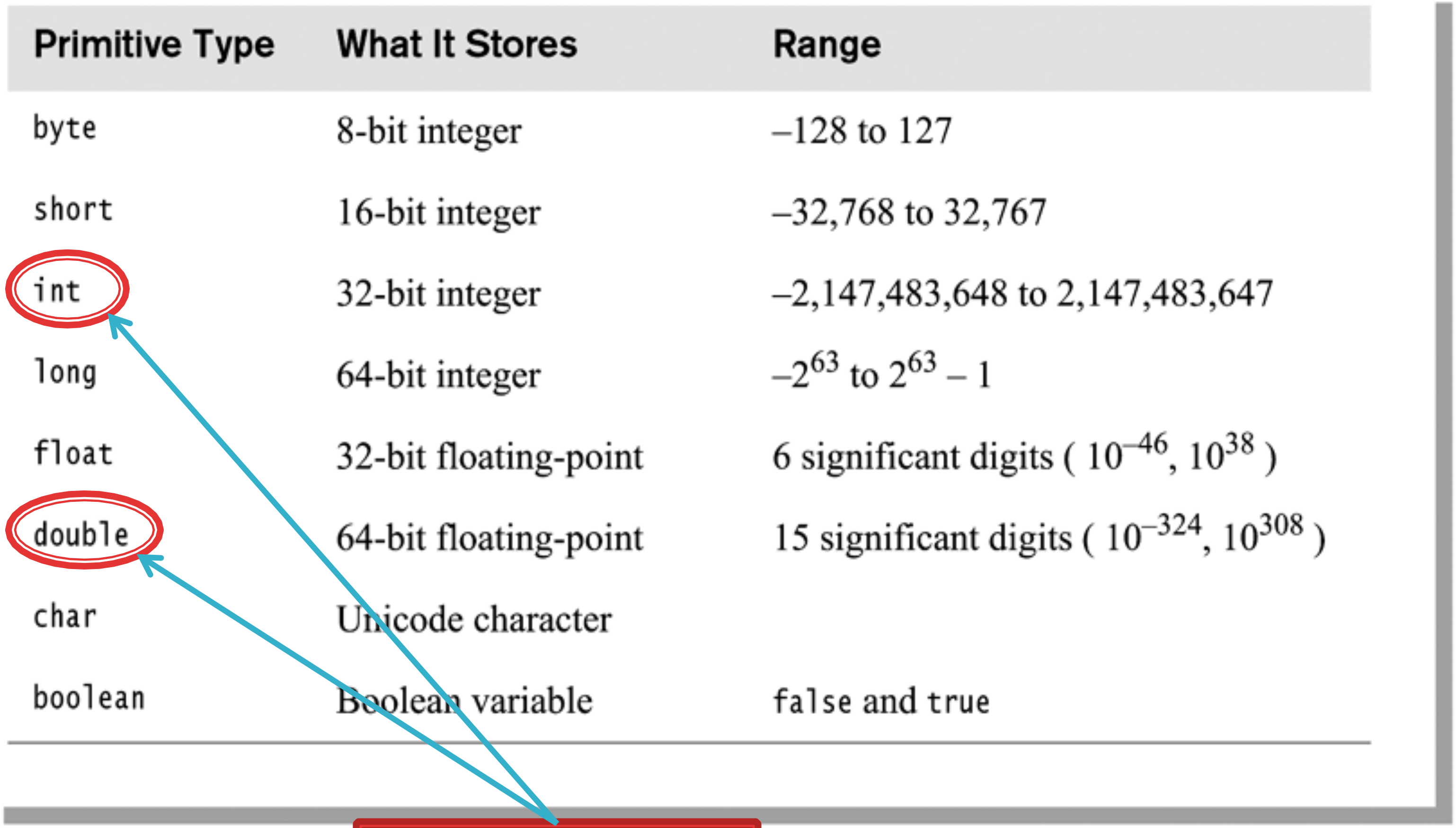
What will I spend my time doing?

- Small programming assignments in class
- Larger programming problems, mostly outside of class
  - Explore the JDK documentation to find the classes and methods that you need
  - Lots of testing and debugging!
  - Reviewing other students' code
- Reading (a lot to read at the beginning; less later)
  - Thinking about exercises in the textbooks
  - Some written exercises, mostly from the textbook
- Discussing the material with other students

Today

- Primitive types
- Loop review
- The String class

Primitive types



Primitive Type	What It Stores	Range
`byte`	8-bit integer	–128 to 127
`short`	16-bit integer	–32,768 to 32,767
`int`	32-bit integer	–2,147,483,648 to 2,147,483,647
`long`	64-bit integer	-2^{63} to $2^{63} - 1$
`float`	32-bit floating-point	6 significant digits (10^{-46}, 10^{38})
`double`	64-bit floating-point	15 significant digits (10^{-324}, 10^{308})
`char`	Unicode character	
`boolean`	Boolean variable	`false` and `true`

figure 1.2

The eight primitive types in Java

Most common number types in Java code

Copyright © 2006 Pearson Addison-Wesley. All rights reserved.

Q1

Exercise

- Work on SomeTypes.java

Review Loops: while & for Loops

- While loop syntax: Similar as Python

```
while (condition) {
  statements
}
```

- For loop syntax: Different from Python

```
for (initialization ; condition ; update) {
      statements
}
```

In both cases, curly braces optional if only one statement in body; but be careful!

Java Loop Examples

- Look at `Investment.java`, `InvestmentTest.java` and `InvestmentRunner.java`
  - Practice using a single while loop
  - Study and run the code, then answer quiz questions

- Do the Rates exercise in the Rates.java file
  - You'll practice using a single for loop in that exercise
  - **Hint**: in printf's format string, use %% to display a single %

Q2–Q3

Sentinel Values: A Loop and a Half

- *Sentinel value*—a special input value not part of the data, used to indicate end of data set
  - `Enter a quiz score, or Q to quit:`

- *A loop and a half*—a loop where the test for termination comes in the **middle** of the loop

- Examples… (on next slide)

Q4–Q5

Two Loop-and-a-half Patterns

```
// Pattern 1

boolean done = false;
while (!done) {
   // do some work

   if (condition) {
      done = true;
   } else {
      // do more work
   }
}
```

The variable *done* here is called a *flag*

```
// Pattern 2

while (true) {
   // do some work

   if (condition) {
        break;
   }

   // do more work
}
```

Q6

Exercise

- Work on UsingStrings.java

Q7

Passing Parameters

- Arguments can be any expression of the “right” type
  - See example…
- What happens if we try to give `substring()` an explicit argument that isn’t a number?
  - How does the compiler know that `rhit.length()` evaluates to a number?
  - What’s the return type of `length()`?

```
String rhit = “Rose-Hulman”;
System.out.println("Rose");
System.out.println(rhit.substring(0, 4));
System.out.println(rhit.substring(0, 2+2));
System.out.println(rhit.substring(0, rhit.length() - 7));
System.out.println("Rose-Hulman".substring(0, 4));
```

Work Time

- Wrap up **Rates and UsingStrings** if you haven’t already, then continue working on TwelveProblems

Q8–Q9, turn in quiz now