CompSci 6
Programming Design and Analysis

September 3, 2009
Prof. Rodger and Prof. Forbes

Announcements

• Reading for next time
  – Chap. 4.1-4.5, Chap 5, Chap 7.1-7.5
  – Reading Quiz due before next class

Top 10 list for surviving in CompSci 6

• 10. Read the Book
• 9. Keep Pizza number handy
• 8. Learn how to spell Rodger
• 7. Ask questions
• 6. Keep working until it is correct

Top 10 list (cont)

• 5. Get the easy points! (reading quizzes, READMEs, etc.)
• 4. Visit your professor, TA and/or UTA
• 3. Read the CompSci 6 Bulletin Board
• 2. Seek help when stuck (1 hour rule)!
• 1. Start programming assignments early!
Estimation

• Square Root:
  – Given a real number $c$ and some error tolerance $\epsilon$
  – Estimate $t$, the square root of $c$

• $\pi$:
  – Estimate $\pi$ with a given number of Monte Carlo trials

While Loops: Square Root

• Q. How might we implement `Math.sqrt()`?
  • A. To compute the square root of $c$:
    – Initialize $t_0 = c$.
    – Repeat until $t_i = c / t_i$, up to desired precision:
      set $t_{i+1}$ to be the average of $t_i$ and $c / t_i$.

<table>
<thead>
<tr>
<th>$t_i$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_0$</td>
<td>$c$</td>
</tr>
<tr>
<td>$t_1$</td>
<td>$\frac{c}{t_0}$</td>
</tr>
<tr>
<td>$t_2$</td>
<td>$\frac{c}{t_0} + \frac{c}{t_1}$</td>
</tr>
<tr>
<td>$t_3$</td>
<td>$\frac{c}{t_1} + \frac{c}{t_2}$</td>
</tr>
<tr>
<td>$t_4$</td>
<td>$\frac{c}{t_2} + \frac{c}{t_3}$</td>
</tr>
<tr>
<td>$t_5$</td>
<td>$\frac{c}{t_3} + \frac{c}{t_4}$</td>
</tr>
</tbody>
</table>

computing the square root of 2

Newton-Raphson Method

• Square root method explained. $f(x) = x^2 - c$ to compute $\sqrt{c}$
  – Goal: find root of function $f(x)$.
  – Start with estimate $t_0 = c$.
  – Draw line tangent to curve at $x = t_i$.
  – Set $t_{i+1}$ to be $x$-coordinate where line hits $x$-axis.
  – Repeat until desired precision.

Buffon Needle Experiment
Needle Position

- Needle length = 1, distance between lines = 2
- Generate random \( y_{low} \) between 0 and 2
- Generate random angle \( \alpha \) between 0 and 180 degrees
- \( y_{high} = y_{low} + \sin(\alpha) \)
- Hit if \( y_{high} \geq 2 \)

Constructing objects/Applying methods

- Class Rectangle in Chapter 2
- Creating a Rectangle object with \( x, y, \) width, and height
  
  \[
  \text{Rectangle box = new Rectangle}(5, 10, 20, 30);
  \]
- Applying Methods
  
  \[
  \text{box.translate}(15, 25); \quad \text{// move the rectangle}
  \]
  
  \[
  \text{System.out.println(“x: “, box.getX()); \quad \text{// print x}
  \]
  
  \[
  \text{System.out.println(“y: “, box.getY()); \quad \text{// print y}
  \]

Parts of a Class

- State
  - Data
- Constructors
  - Initialize state when object is created
- Accessor methods
  - Accessing data
- Mutator methods
  - Modify data – change the state

Class Example

- Needle class – Needle.java
  - Defines state and behavior of Needle
  - Keeps track of the number of times needle hits the line
  - Use drop() method to simulate dropping needle
- java.util.Random class in Java library
  - nextDouble() generates pseudo-random numbers in \([0,1]\)
```java
import java.util.Random;

public class Needle {
    public Needle() {
        hits = 0;
        tries = 0;
        generator = new Random();
    }

    public void drop() {
        double ylow = 2 * generator.nextDouble();
        double angle = 180 * generator.nextDouble();

        double yhigh = ylow + Math.sin(Math.toRadians(angle));
        if (yhigh >= 2) myHits++;
        tries++;
    }

    public int getHits() {
        return myHits;
    }

    public int getTries() {
        return myTries;
    }

    private Random myGenerator;
    private int myHits;
    private int myTries;
}
```

**Classwork Today – Loops/Classes**

- Snarf the `classwork/04_loops_cps006_fall09` project
- Complete Sqrt
  - Finish `estimate` method
  - Print results
- Complete Needle
  - Finish `main` method
  - Print results
- Classwork handout has all the details
- Submit under assignment name `classSep03`

**Intended Output:**

Tries = 10000, Tries / Hits = 3.08928
Tries = 1000000, Tries / Hits = 3.14204