Over and Over Again

GEEN163

“There is no harm in repeating a good thing.”

Plato

Homework

- A programming assignment has been posted on Blackboard
- You have to convert three flowcharts into programs
- Upload the three .java files to Blackboard by midnight on Friday, October 11, 2013

Summations in Mathematics

- In mathematics you can specify a sum as

$$sum = \sum_{i=1}^{n} \frac{1}{2i}$$

- which is equivalent to

$$sum = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2n}$$

Summations in Java

- We can write the same sum in Java

$$sum = \sum_{i=1}^{n} \frac{1}{2i}$$

```
double sum = 0.0;
int i = 1, n = something;
while (i <= n) {
        sum += 1.0 / (2.0 * i);
        i++;
}
```

Looping Structures

- Java provides three different ways to create a loop

  `while` loops

  `do while` loops

  `for` loops

- All of them cause the program to repeat a set of statements

do while loops

- The do while loops is very similar to the while loop except that the loop condition is checked at the end of the loop

```
do {
    sum += 1.0 / (2.0 * i);
    i++;
} while ( i <= n );
```

do while Syntax

- When a do while loop gets to the end, it checks the logical expression. If true, it repeats the loop

```
do {
    // loop body
} while (logical expression);
```

Always Once

- A do while loop is always executed at least once

```
int hen = 5;
while ( hen < 4 ) {
  hen = hen * hen; // never executed
}

do {
  hen = hen * hen; // executed once
} while( hen < 4 );
```

Not Repeat Until

- When the logical expression of a **`do while`** loop is true, the loop repeats
- Write the logical expression to express when the loop repeats
- Some students incorrectly think a **`do while`** is a repeat until where the logical expression tells you when to stop

Invert Repeat Logic

- In the loop body compute a new term
- Repeat until the term is **less than** 0.001

```
do {
    term = term * x / i;
    sum += term;
    i++;
} while (term >= 0.001);
```

Watch the semicolon

- The following loop is wrong:

```
int i=0;
while (i < 10);
{
      System.out.println("i is " + i);
      i++;
}
```

Logic Error

- With the do loop, the semicolon is required:

```
int i=0;
do {
      System.out.println("i is " + i);
      i++;
} while (i<10);
```

Correct

What is displayed?

```
int dog = 3, cat = 5;
while ( cat < dog ) {
   System.out.print( dog );
   dog++;
}
```

A. 3

B. 3 4

C. 3 4 5

D. nothing

Now what is displayed?

```
int dog = 3, cat = 5;
do {
    System.out.print( dog );
    dog++;
while ( cat < dog );
}
```

A. 3

B. 3 4

C. 3 4 5

D. nothing

Thinking about programs

- If a program has to do something many times, it will need a loop
  - How many times will the program loop?
- The parts of the program that are not repeated will be outside the loop
- If a program does something different some of the time, the program will have an **`if`** statement

Inputs and Outputs

- What is the program expected to display?
- What data will the program need to get to produce the results?
  - What data types will hold the information

What is displayed?

```
int tortoise = 3, hare = 0;
do {
    hare += tortoise;
    tortoise--;
} while (tortoise > 0);
System.out.println(tortoise+" "+hare);
```

1. 3 0
2. 1 5
3. 0 6
4. none of the above

Variables Needed

- When you write a program, consider the variables necessary
- You will probably need a variable to hold each input
- You will probably need a variable to hold the result of a calculation

Patterns

- We have learned a few programming patterns that perform common tasks
  - read until a value
  - repeat a fixed number of times
- These patterns are not just one Java statement, but are several Java statements
- These patterns appear in many programs

Priming Read

```
System.out.print("enter number >");
cat = keyboard.nextInt();
while (cat != endValue ) {
        // do something with cat
        System.out.print("enter number >");
        cat = keyboard.nextInt();
}
```

Same

Repeat a Fixed Number of Times

- Starting with one

```
int counter = 1;
while (counter <= maxValue) {
        // do something
        counter++;
}
```

Repeat a Fixed Number of Times

- Starting with zero

```
int counter = 0;
while (counter < maxValue) {
        // do something
        counter++;
}
```

Sample Program

Consider a program that calculates the cost of purchasing some coffee. Input the number of pounds of coffee purchased and the price per pound of the coffee. Display the total cost which includes 7% sales tax. Repeat until the amount of coffee is zero or negative.

What are the inputs to the program?

A. pounds of coffee
B. pounds & price
C. pounds, price & cost
D. pounds, price, cost & tax

What results should be displayed?

A. total cost
B. total cost and tax
C. pounds, price, tax & cost
D. None of the above

Likely Variables Required

- Input values
  - pounds of coffee
  - price of coffee
- Output value
  - total cost

Program Outline

```
import java.util.Scanner;
public class CoffeeCost {
    public static void main( String[] args) {
        Scanner keyboard = new Scanner(System.in);
         // repeat
            // read input
            // calculate cost with tax
            // display results
    }
}
```

Declare the variables

```
import java.util.Scanner;
public class CoffeeCost {
    public static void main( String[] args) {
        Scanner keyboard = new Scanner(System.in);
        double pounds, price, cost;
       //  repeat
         // read input
         // calculate cost with tax
         // display results
    }
}
```

Write the input section

Input the number of pounds of coffee purchased and the price per pound of the coffee

Possible Solution

```
System.out.println("Enter the pounds of coffee");
pounds = keyboard.nextDouble();
System.out.println("Enter the price per pound");
price = keyboard.nextDouble();
```

Write the cost calculation and display the result

- Display the total cost including 7% sales tax.

Possible Solution

```
cost = pounds * price * 1.07;
System.out.println("The coffee costs "+
                                    cost );
```

Adding the Loop

- The program is to repeat until the user enters zero or negative pounds
- There is no use asking the price if the pounds are zero or negative
- A **“priming read”** will help

Possible Solution

```
pounds = keyboard.nextDouble();
while (pounds > 0.0) {
        System.out.println(
                        "Enter the price per pound");
        price = keyboard.nextDouble();
        // rest of the program
        pounds = keyboard.nextDouble();
}
```

Complete Program

```
public class CoffeeCost {
   public static void main( String[] args) {
      java.util.Scanner keyboard = new
                           java.util.Scanner(System.in);
      double pounds, price, cost;
      // read input
      System.out.println("Enter the pounds of coffee");
      pounds = keyboard.nextDouble();
      while (pounds > 0.0) {
         System.out.println("Enter the price per pound");
         price = keyboard.nextDouble();
         // calculate cost with tax
         cost = pounds * price * 1.07;
         // display results
         System.out.println("The coffee costs "+ cost );
         System.out.println("Enter the pounds of coffee");
         pounds = keyboard.nextDouble();
      }
   }
}
```

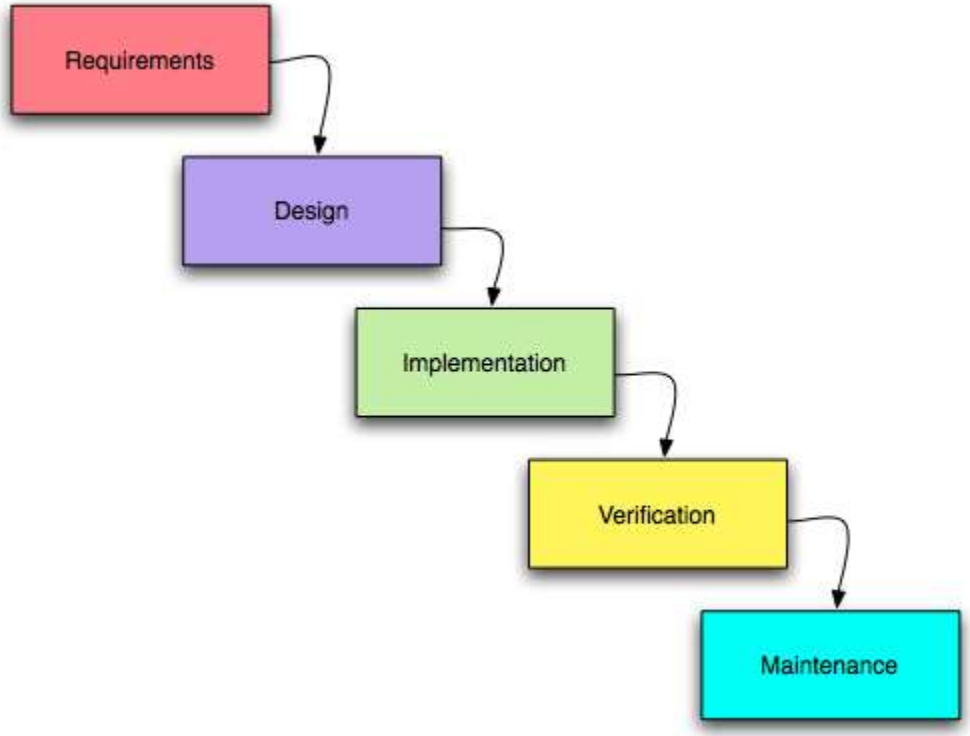
Software Engineering Steps

- **Requirements** – Define exactly what the program is supposed to do from the user's point of view
- **Design** – How will the program run
- **Implementation** – Write the Java
- **Verification** – Test the program looking for any errors
- **Maintenance** – Fix the errors you missed the first time

Waterfall Model

- The waterfall model is one software engineering technique for creating programs

- There are several other software engineering processes



Homework

- Write three simple programs using while loops and if statements
- Flowcharts are provided to explain how to write the program
- Due midnight on Friday, October 11