COMP 110-001
More About Loops

Yi Hong

May 21, 2015

Review

- Q1: What is the output of the following statements?

```
int count = 0;
int sum = 0;
while(count <= 100) {
     count = count + 2;
     if(count > 100) break;
     if(count % 3 != 0) continue;
     sum = sum + count;
}
System.out.println(sum);
```

  - The sum of multiples of 6 within [0, 100]

Review

- Q2: How many iterations?

`for (count = 1; count < 10; count++)`	9 times
`for (count = 1; count <= 10; count++)`	10 times
`for (count = 0; count <= 10; count++)`	11 times
`for (count = 0; count < 10; count++)`	10 times
`for (count = 1; count < 10; count+=2)`	5 times

Today

- How to write loops
- Bugs in loops

Loop Body

```
count = 1;
while (count <= num)
{
    System.out.print(count +  “,  “);
    count++;
}
```

- Repeated code
- Write pseudocode and turn repeated statements into loops

Pseudocode for a Loop

- Get user input
- sum = sum + input
- Get user input
- sum = sum + input
- Get user input
- sum = sum + input
- Average sum

Repeated statements in pseudocode become your loop

Body of The Loop

- Get user input
- sum = sum + input

Initializing Statements

sum = sum + input

- Variables in a loop must be initialized (set to a value) before the loop
- What is initialization of sum?
- What if we want to compute the product?
  - sum = sum * input

Ending a Loop

- If you know number of loop iterations
  - Count-controlled loops (the for loop)
- User controlled ending
  - Ask-before-iterating
  - Sentinel value
- Booleans

Count-Controlled Loops

```
for(count = 0; count < iterations; count++)
{
   System.out.print(“I have iterated ” +
   (count + 1) +  “times\n”  );
}
```

Ask-Before-Iterating

```
do
{
   //do stuff in your code here
   System.out.print("Continue? yes/no");
   answer = keyboard.next();
} while(answer.equalsIgnoreCase("yes"));
```

Sentinel Value

- Signal end of input

```
System.out.println("enter a negative number to end the loop");
int next = keyboard.nextInt();
sum  = 0;
while (next >= 0)
{
    sum = sum + next;
    System.out.println("enter a number");
    next = keyboard.nextInt();
}
```

Booleans

```
int next, sum = 0;
boolean numbersLeft = true;
Scanner keyboard = new Scanner(System.in);
while (numbersLeft)
{
    next = keyboard.nextInt();
    if (next < 0)
       numbersLeft = false;
    else
       sum = sum + next;
}
System.out.println(“The sum is ” + sum);
```

Write Code

- Give a Java loop statement that will set the variable result equal to 2^5
- Write a program that maintains the balance of an account
  - Ask for a balance-update from user in each iteration
    - Positive value: deposit
    - Negative value: withdraw
  - If the balance-update is 0 or the balance goes below 0, exit from loop and print out the remaining balance

Nested Loop

- What does the following statements do?

```
for(int i = 0; i < 10; i++)
{
    for(int j = 0; j < 10; j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

Nested Loop

- What does the following statements do?

```
for(int i = 0; i < 10; i++)
{
    for(int j = 0; j < i; j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

```
*
**
***
****
*****
******
*******
********
*********
```

Bugs

- Problems in a program that prevent correct execution

- Two most common mistakes in loops
  - Off-by-one errors
  - Infinite Loops!!!!!!

Off-by-one errors

- Loop repeats one more or less time
  - E.g.: If you want a program to repeat 10 times
    - for (count = 1; count < 10; count++);
      - Loop 9 times
    - for (count = 1; count <= 10; count++);
      - Loop 10 times
    - for (count = 0; count < 11; count++);
      - Loop 11 times
    - for (count = 0; count < 10; count++);
      - Loop 10 times

Infinite Loops

- A loop which repeats without ever ending is called an *infinite loop*
- If the controlling boolean expression never becomes false, a loop will repeat without ending

Infinite Loops

```
count = 1;
while (count <= num)
{
   System.out.print(count +   “,   “);
   //count++;
}
```

Infinite Loops

```
count = 1;
while (count <= num);
{
   System.out.print(count +   “,   “);
   count++;
}
```

Infinite Loops

```
int count;
int num = 1;
// initializing action; boolean expression; update action
for (count = 1; count >= num; count++)
{
   System.out.print(count +   “,   “);
   num = count;
   num++;
}
```

Finding Errors

- Error checking
  - System.out.print(variable);
  - Run on simple input
- Debug (Required only for CS students or who are interested in debugging)
  - Eclipse: breakpoint + variable watch

Try It Yourself

- Let's print out a Multiplication Table

```
for(int i = 1; i < 10; i++)
{
    for(int j = 1; j <= i; j++)
    {
        System.out.print(i+"*"+j+"="+(i*j)+"\t");
    }
    System.out.println();
}
```

- Output

```
1*1=1
2*1=2   2*2=4
3*1=3   3*2=6   3*3=9
4*1=4   4*2=8   4*3=12  4*4=16
5*1=5   5*2=10  5*3=15  5*4=20  5*5=25
6*1=6   6*2=12  6*3=18  6*4=24  6*5=30  6*6=36
7*1=7   7*2=14  7*3=21  7*4=28  7*5=35  7*6=42  7*7=49
8*1=8   8*2=16  8*3=24  8*4=32  8*5=40  8*6=48  8*7=56  8*8=64
9*1=9   9*2=18  9*3=27  9*4=36  9*5=45  9*6=54  9*7=63  9*8=72  9*9=81
```

Next Class

- Classes
- Reading assignment: Chapter 5.1