We Continue

GEEN163

"I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone."

Bjarne Stroustrup

creator of C++

Exam

- The second exam will be in lecture on Wednesday, October 23
- The exam will cover everything since the first exam
    - If statements
    - Loops
    - Files
- A sample exam is available on Blackboard under course materials
- You may have one 8½ x 11" page of notes

Recitation Quiz

There will be a quiz in recitation on

- Thursday or Friday for the 10:00 lecture
- Monday or Tuesday for the 9:00 lecture

Lab Quiz

- There will be a quiz in lab next week
- The quiz will cover loops and ifs
- You will be required to complete a program on your own without the help of others
- You may use your book, notes and the web

for and while loop equivalence

```
for ( initialization ;  test; update )  {
        statements to repeat
}
```

is the same as

```
initialization;
while (test) {
        statements to repeat
        update;
}
```

do while version

- To read 7 integers with a do while loop

```
Scanner keys = new Scanner(System.in);
int  sum = 0, count;
count = 0;
do {
  sum = sum + keys.nextInteger();
  count++;
} while(count < 7);
```

while version

- To read 7 integers with a while loop

```
Scanner keys = new Scanner(System.in);
int  sum = 0, count;
count = 0;
while(count < 7){
  sum = sum + keys.nextInteger();
  count++;
}
```

for version

- To read 7 integers with a for loop

```
Scanner keys = new Scanner(System.in);
int  sum = 0, count;
for (count = 0; count < 7; count++){
  sum = sum + keys.nextInteger();
}
```

Declaring the Loop Counter

- In a for loop, you can declare the variable that is used as the loop counter
- The loop counter can only be used in the loop

```
for (int bull = 0; bull < 47; bull ++) {
        // loop body
        // the loop counter, bull, can be used here
}
// You cannot use bull after the loop
```

What does this print?

```
int dog = 1;
for (int cat = 3; cat < 5; cat++) {
    dog = dog + cat;
}
System.out.print( dog );
```

A. 1

B. 8

C. 13

D. none of the above

Loops in Loops

- Sometimes a program will need one loop in the body of another loop

```
int sum = 0, cat = 5, dog = 3;
for (int i = 0; i < cat; i++) {
    for (int k = 0; k < dog; k++) {
        sum += i * k;
    }
}
```

- Note that the loop counter variables (i and k) must be different

Long Running Programs

- If a loop repeats **n** times is inside a loop that repeats **m** times, the total number of iterations of the loop is **m * n**
- If **n** and **m** are large, it may take a very long time for the program to complete
- Loops inside loops inside loops inside loops can take a long time to run

Selecting from Multiple Choices

- The if - else if statements are a way of selecting from multiple choices

```
if ( x ) {
    // if x is true
} else if ( y ) {
    // if x is false and y is true
} else if ( z ) {
    // if x and y are false and z is true
} else {
    // if x, y and z are false
}
```

Write Java with your Team

- Set the String variable weather to "calm" if wind is less than 3.0, "strong" if wind is greater than 24.0 and "breeze" otherwise

```
double wind = something;
```

Possible Solution

- Set the String variable weather to "calm" if wind is less than 3.0, "strong" if wind is greater than 25.0 and "breeze" otherwise

```
double wind = something;
if (wind < 3.0) {
        weather = "calm";
} else if (wind > 25.0) {
        weather = "strong";
} else {
        weather = "breeze";
}
```

What is weather when wind is 2.5?

```
double wind = something;
if (wind < 3.0) {
        weather = "calm";
}
if (wind > 25.0) {
        weather = "strong";
} else {
        weather = "breeze";
}
```

A. calm
B. strong
C. breeze
D. 3.0
E. 25.0

Scanner with Files

- When you create an object of the Scanner class, you can specify a File object instead of System.in

```
java.io.File  elephant = new java.io.File("mydata.txt");
java.util.Scanner pachyderm =
                              new java.util.Scanner( elephant );
        // read a number from the file
int mouse = pachyderm.nextInt();
```

Complete this program to average the numbers in myData.txt

```
public class SumFile {
   public static void main(String[] args) throws
                  java.io.FileNotFoundException {
      java.io.File dog =
                  new java.io.File("myData.txt");
      java.util.Scanner cat =
                  new java.util.Scanner( dog );
```

Possible Solution

```
public class SumFile {
   public static void main(String[] args) throws
                  java.io.FileNotFoundException {
      java.io.File dog =
                  new java.io.File("myData.txt");
      java.util.Scanner cat =
                  new java.util.Scanner( dog );

      double sum = 0.0;     // sum of numbers
      int    count = 0;     // count of numbers
      while (cat.hasNextDouble()) {
         sum += cat.nextDouble();
         count++;
      }
      System.out.println("Average is"+ sum/count);
   }
```

break statement

- The break statement stops a loop
- The break statement can be used in for loops, while loops, do-while loops and switch statements
- A break statement causes the loop to stop looping and starts executing after the loop

break example

```
int sum = 0;
for (int i = 0; i < 5; i++) {
        sum = sum + i * i;
        if ( sum > 10)
                break;
}
System.out.println(sum);
```

i	sum
0	0
1	1
2	5
3	14

// prints 14

Loops in Loops

- If you have loops inside other loops, a break statement will stop the looping of the inner most loop where the break is executed
- Outer loops will not be effected by a break in an inner loop.

Break in a Loop in a Loop

```
public class BreakSum {
 public static void main(String[] args) {
   int sum = 0;
   for (int i = 1; i < 3; i++ ) {
     for (int k = 1; k < 3; k++ ) {
       System.out.println("i:"+ i + "  k:"+ k );
       sum = sum + k;
       if (sum > 2 ) break;
       System.out.println( "sum: "+sum );
     }
   }
  System.out.println( "final sum: "+sum );
 }
}
```

```
i:1  k:1
sum: 1
i:1  k:2
i:2  k:1
final sum:4
```

What is Displayed?

```
int apple = 1;
for(int pear = 1; pear<4; pear++) {
   apple++;
   if (apple > 4 ) {
     break;
   }
   apple += pear;
}
System.out.println(apple);
```

A. 1

B. 4

C. 6

D. 7

E. 10

Labeled Statements

- You can put a label before any statement
- Labels start the Java statement and are separated from the statement by a colon
- Label names follow the usual rules

```
fred : for (int i = 0; i < 100; i++) {

mary : while ( dog < 13 ) {
```

break to a label

- To avoid problems with loops in loops, you can put a label on the loop and then specify the label on the break

```
fred : for (int i = 0; i < 13; i++ ) {
                if (i * 2 > 20 )
                        break fred;
        }
```

- When you have loops in loops, this specifies exactly which loop you are terminating

break Labels

- The label on a break statement must indicate a loop statement (for, do or while) that encompasses the break

```
tom: for (int i = 0; i < 47; i++ ) {
        sum += i * i;
        if (sum > 100)  break george; // ERROR
}
george: System.out.println("after loop");
```

Nested Loop Break

```
int cow, bull, sum = 0;
daisy: for (cow = 1; cow <= 3; cow++ ) {
    ferdinand: for (bull = 1; bull <= 5; bull++ ) {
        sum += cow * bull;
        if (sum > 25) break daisy; // ends both loops
    }
}
```

continue Statement

- The **`continue`** statement makes it go the beginning of the loop and start the next iteration

```
for (int cow = 0; cow < 100; cow++) {
        // something
        if ( cow < 47 || cow > 53 )
                continue;
        // some other thing
        System.out.println( cow );
}
```

This prints 47 to 53

Continue in a Loop in a Loop

```
public class BreakSum {
 public static void main(String[] args) {
    int sum = 0;
    for (int i = 1; i < 3; i++ ) {
      for (int k = 1; k < 3; k++ ) {
        System.out.println("i:"+ i + "  k:"+ k );
        sum = sum + k;
        if (sum > 2 ) continue;
        System.out.println( "sum: "+sum );
        // other stuff
      }
    }
    System.out.println( "final sum: "+sum );
 }
}
```

```
i:1  k:1
sum: 1
i:1  k:2
i:2  k:1
i:2  k:2
final sum:6
```

What is the final value of apple?

```
int apple = 1;
for(int pear = 1; pear<4; pear++) {
   apple++;
   if (apple > 4 ) {
     continue;
   }
   apple += pear;
}
System.out.println(apple);
```

A. 1

B. 4

C. 5

D. 8

E. none of the above

Labels on Continue

- Just like the break statement, you can put labels on a continue statement
- The label must be on a surrounding loop, either for, while or do while

Labels are Recommended

- Many Java programs do not use labels on the break or continue statements
- Many Java programs have errors due to misunderstood break and continue statements
- The use of labels makes your intentions clear
- The compiler will tell you if you are doing something wrong

Software Failure

Due to a programmer's misunderstanding of how a break statement works, a NASA probe crashed into the surface of Venus

Modify this to average the first 100 numbers in myData.txt

```
public class SumFile {
   public static void main(String[] args) throws
                  java.io.FileNotFoundException {
      java.io.File dog =
                  new java.io.File("myData.txt");
      java.util.Scanner cat =
                  new java.util.Scanner( dog );

      double sum = 0.0;     // sum of numbers
      int    count = 0;     // count of numbers
      while (cat.hasNextDouble()) {
         sum += cat.nextDouble();
         count++;
      }
      System.out.println("Average is "+sum/count);
   }
}
```

Possible Solution

```
public class SumFile {
   public static void main(String[] args) throws
                  java.io.FileNotFoundException {
      java.io.File dog =
                  new java.io.File("myData.txt");
      java.util.Scanner cat =
                  new java.util.Scanner( dog );

      double sum = 0.0;     // sum of numbers
      int    count = 0;     // count of numbers
      while (cat.hasNextDouble() && count < 100) {
         sum += cat.nextDouble();
         count++;
      }
      System.out.println("Average is "+sum/count);
   }
}
```

Possible Solution

```
public class SumFile {
   public static void main(String[] args) throws
                     java.io.FileNotFoundException {
      java.io.File dog =
                     new java.io.File("myData.txt");
      java.util.Scanner cat =
                     new java.util.Scanner( dog );

      double sum = 0.0; // sum of numbers
      int    count = 0; // count of numbers
      readLoop: while (cat.hasNextDouble()) {
         sum += cat.nextDouble();
         count++;
         if (count == 100) {
            break readLoop;
         }
      }
      System.out.println("Average is"+sum/count);
   }
}
```

Lab Quiz

- There will be a quiz in lab next week
- The quiz will cover loops and ifs
- You will be required to complete a program on your own without the help of others
- You may use your book, notes and the web

Exam

- The second exam will be in lecture on Wednesday, October 23
- The exam will cover everything since the first exam
    - If statements
    - Loops
    - Files
- A sample exam is available on Blackboard under course materials
- You may have one 8½ x 11" page of notes

Recitation Quiz

There will be a quiz in recitation on

- Thursday or Friday for the 10:00 lecture
- Monday or Tuesday for the 9:00 lecture

Programming Assignment

- A reading level program is due Friday at midnight