

More on Loops

GEEN163

*“Beware of bugs in the above code;
I have only proved it correct, not
tried it.”*

Donald Knuth

Homework

- The GUI number guessing game programming assignment has been posted on Blackboard
- Homework is due **today** at midnight

boolean variables

- A boolean variable can only have the values **true** or **false**.

```
boolean rabbit, err = false;  
rabbit = true;  
if (a > 5) {  
    err = true;  
}
```

Logical Equations

- A boolean variable can be set to the result of a logical expression.

```
int x=3, y=5, z=7;
```

```
boolean bat, bird;
```

```
bird = x > y;
```

```
bat = (x != y) && (z > y);
```

- The expression is evaluated once and the boolean variable is then set to true or false.
- Changing **x**, **y** or **z** will not change the value of **bat** after the above equations.

boolean variables in IF

- You can use a boolean variable in an if statement without a comparison.

```
boolean    problem = false;
```

```
...
```

```
problem = true;
```

```
...
```

```
if ( problem ) {
```

```
    System.out.println("look out");
```

```
}
```

boolean Methods

- A method can return a boolean (true or false) value

```
boolean close(double cat, double dog) {  
    if ( Math.abs(cat - dog) < 0.01 ) {  
        return true;  
    }  
    return false;  
}
```

A Shorter boolean Method

- A comparison results in a true or false value which can be returned

```
boolean close(double cat, double dog) {  
    return Math.abs(cat - dog) < 0.01;  
}
```


Using boolean Methods

- A boolean method can be used in an if

```
double cobra = 0.666, mamba = 2.0 / 3.0;  
if ( close( cobra, mamba ) )  
    System.out.println("same");
```

- Other boolean methods include **equals**, **equalsIgnoreCase** and others

Input Validation

```
int input;
boolean good = false;
while ( !good ) {
    System.out.println("Enter a number from 1-5");
    input = keyboard.nextInt();
    if (input < 1 || input > 5) {
        System.out.println("Pay attention!!");
    } else {
        good = true;
    }
}
```

boolean variables hold logical values

- A boolean variable can only hold the value **true** or **false**
- **true** and **false** are keywords, not “strings”, but they will print as strings

```
int cat = 3, dog = 5;
```

```
boolean fish = cat < dog;
```

```
System.out.println("fish is "+ fish);
```

- will print **fish is true**

What is the final value of goat?

```
int cat = 11, dog = 5, cow = 7, goat = 1;  
boolean squid;  
squid = dog + cow == cat + 1;  
if (squid )  
    goat = 8;  
else  
    goat = 5;
```

A. 1

B. 5


C. 8

D. 12

E. none of the above

Caution

The following loop is wrong:

```
int i=0;
while (i < 10);  Logic Error
{
    System.out.println("i is " + i);
    i = i + 1;
}
```

Looping a Specified Number of Times

- Frequently you may want your program to loop **n** times.

```
int month = 1, n = 10;
```

```
double principle= 10000.0;
```

```
while (month <= n) {
```

```
    principle = principle * 1.05;
```

```
    month++;
```

```
}
```

```
System.out.println("The value is "+ principle);
```

Looping a Specified Number of Times

- Computer scientist often count starting at zero.

```
int month = 0, n = 10;
```

```
double principle= 10000.0;
```

```
while (month < n) {
```

```
    principle = principle * 1.05;
```

```
    month++;
```

```
}
```

```
System.out.println("The value is "+ principle);
```

What is displayed?

```
int frog = 0;  
while ( frog < 4 ) {  
    frog++;  
    System.out.print(frog);  
}
```

- A. 0 1 2 3 4
- B. 1 2 3 4
- C. 0 1 2 3
- D. 1 2 3
- E. none of the above

Write a Loop in your Teams

- Write a program segment to read and display integer numbers from a Scanner object named keyboard until you read the number 2
Do not display the number 2.

Possible Solution

```
int cobra;  
cobra = keyboard.nextInt();  
while (cobra != 2) {  
    System.out.println(cobra);  
    cobra = keyboard.nextInt();  
}
```

Modify your program

- Change your program to only write the numbers that are greater than 7
- The program should still continue until it reads a 2

Possible Solution

```
int cobra;  
cobra = keyboard.nextInt();  
while (cobra != 2) {  
    if (cobra > 7) {  
        System.out.println(cobra);  
    }  
    cobra = keyboard.nextInt();  
}
```

Proper Format

- Both the true part and the false part of an if statement should be indented
- The body of a loop should be indented
- ifs in ifs should be doubly indented

What is displayed?

```
int dog = 2, cat = 0;  
while ( dog < 6 ) {  
    dog = dog + cat;  
    cat = cat + 2;  
}  
System.out.println(dog);
```

- A. 2
- B. 4
- C. 6
- D. 8
- E. none of the above

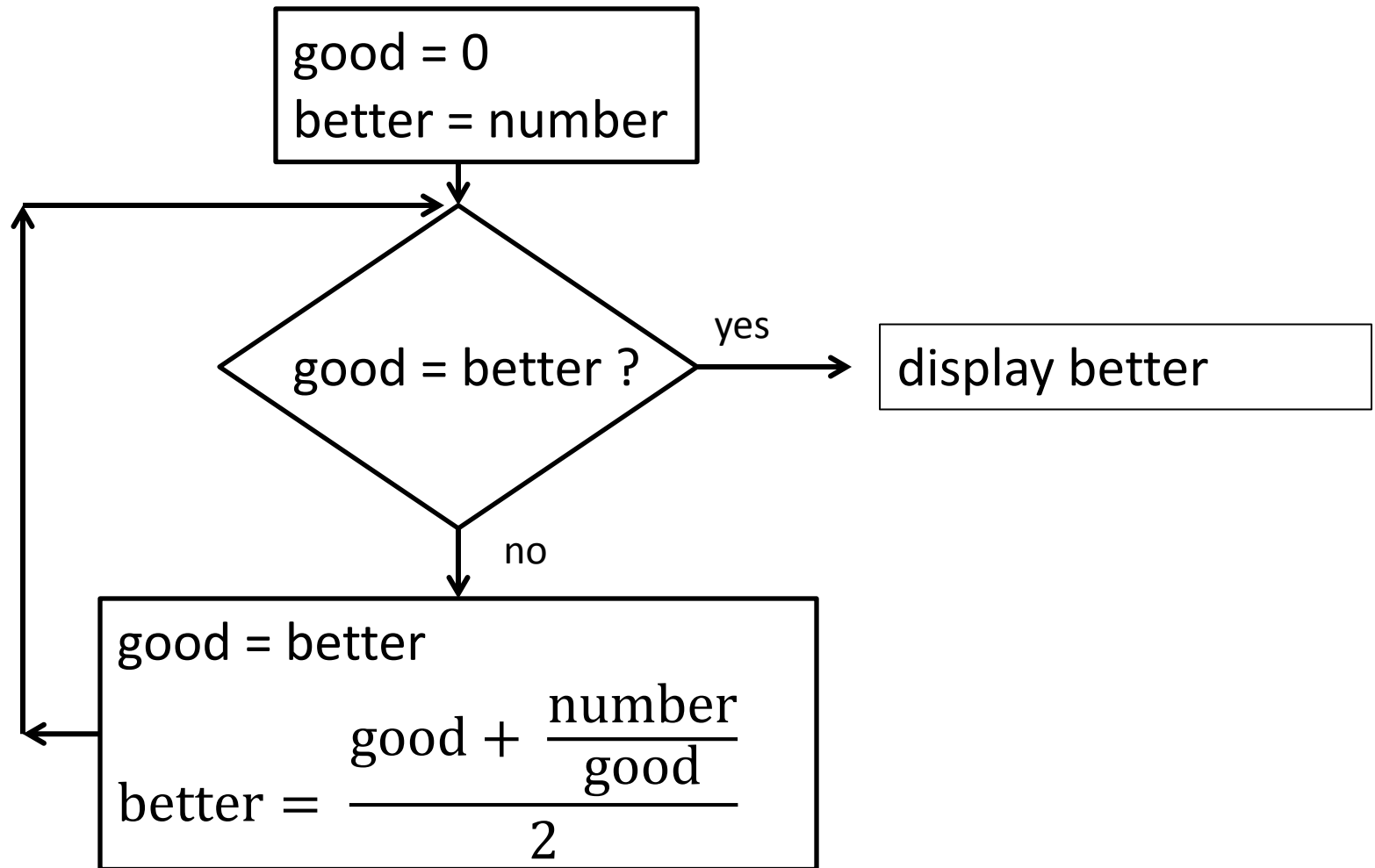
Possible Solution

```
int value, num = 1;
System.out.print("Enter a number >");
value = keyboard.nextInt();
while (num <= value) {
    System.out.println( num );
    num++;
}
```

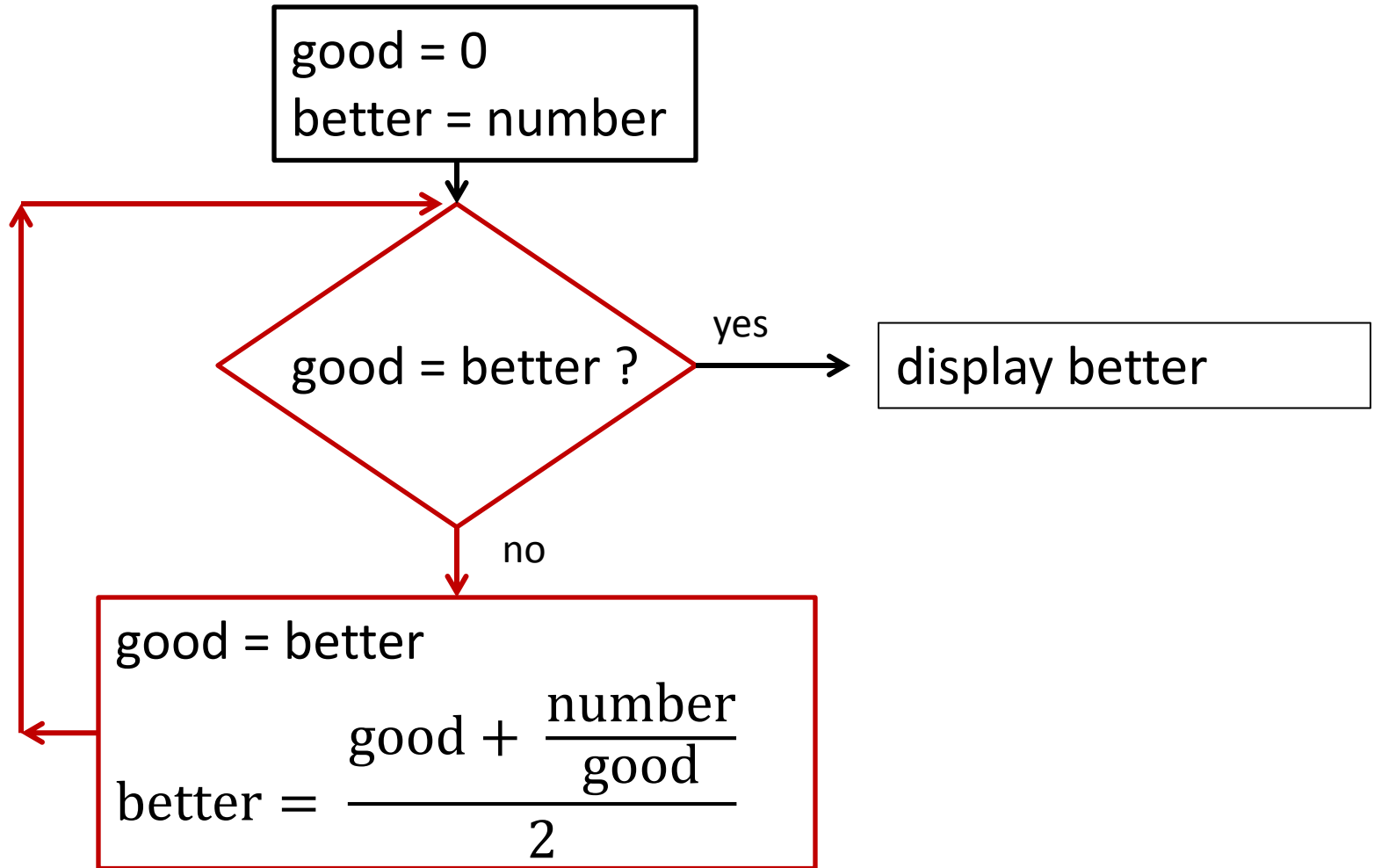
Thinking about programs

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

Write this in Java with your team



Decision box is a while loop because it loops back



Possible Solution

```
int number, good, better;
number = keyboard.nextInt();
good = 0;
better = number;
while (good != better) {
    good = better;
    better = (good + number/good) / 2;
}
System.out.println( better );
```

End of Input

- The `java.util.Scanner` class has boolean methods `.hasNextInt()` and `.hasNextDouble()`
- The methods return `true` if there is another number to read
- The methods return `false` at the “End of File”

```
while ( keyboard.hasNextInt() ) {  
    aardvark = keyboard.nextInt();  
}
```

Write with your Team

- Write a Java segment to read a value and display the numbers from 1 to the value

Enter a number >5

1 2 3 4 5

Possible Solution

```
int value, num = 1;
System.out.print("Enter a number");
value = keyboard.nextInt();
while (num <= value) {
    System.out.print( num );
    num++;
}
```