Program Logic to Java

GEEN163

“In theory, theory and practice are the same. In practice, they're not.”

Yogi Berra

Founders' Day

- The Founders' Day Convocation is Thursday, March 21 from 10:00 – 12:00 in the Harrison Auditorium
- Classes are suspended during that time

- The lab quiz will be next week instead of this week

Exam

- The second exam will be in lecture on Friday, March 22
- The exam will cover everything since the first exam
    - If statements
    - Loops
    - Files
- A sample exam is available on Blackboard under course materials

Thinking About Programs

- If you do not know how to do something, it is very difficult to explain to someone else how to do it
- If you do not know how to solve a problem, it is very difficult to write a program to do it
- It is very useful to think about what a program must do and in what order the steps must be taken

Logical Ordering

- Some things must happen before others
- If you are going to read a number and display it, the read must come before the display

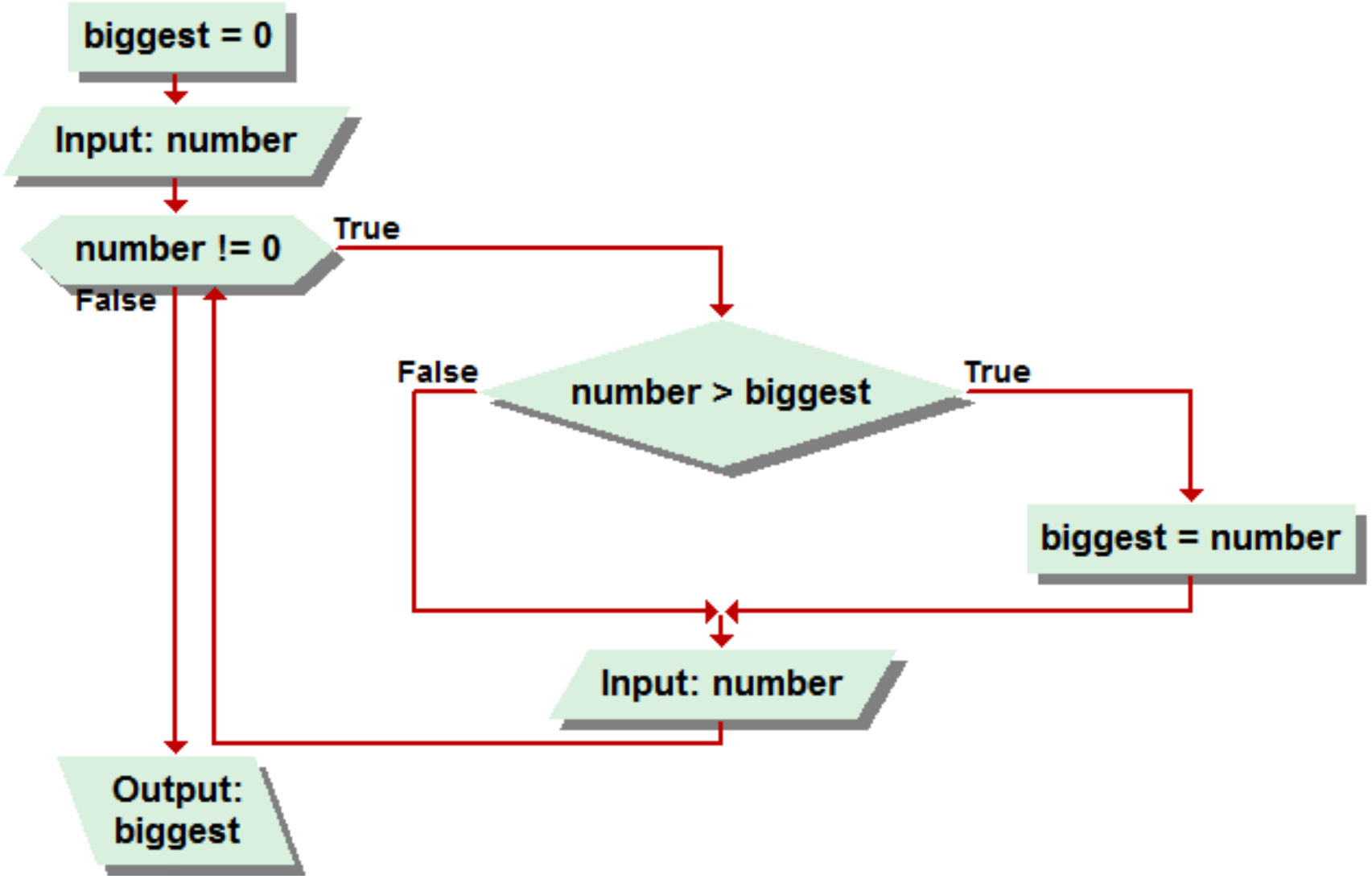
Variables Needed

- 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

Loops and Ifs

- 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

What variables are needed?



Solution

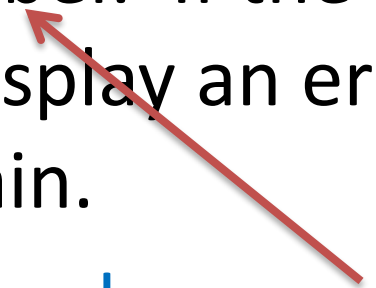
- The program uses two variables
- biggest
- number

Problem Description

- Read a positive number. If the number is not greater than zero, display an error message and ask the user again.

Problem Description

- Read a positive number. If the number is not greater than zero, display an error message and ask the user again.
- This program only needs one variable
- Since we have not been told it is a whole number, it should be defined as a double

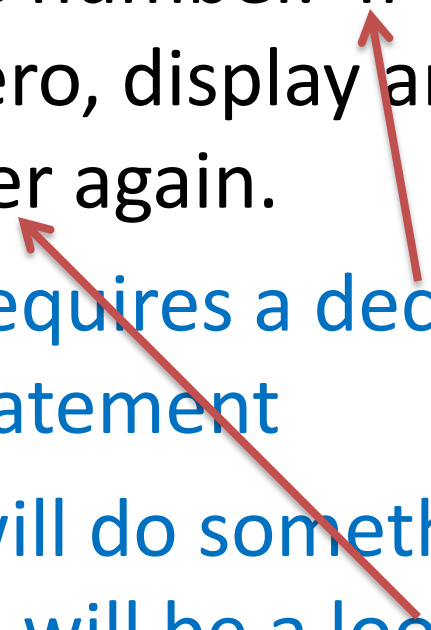


Java Implementation

```
double number;
```

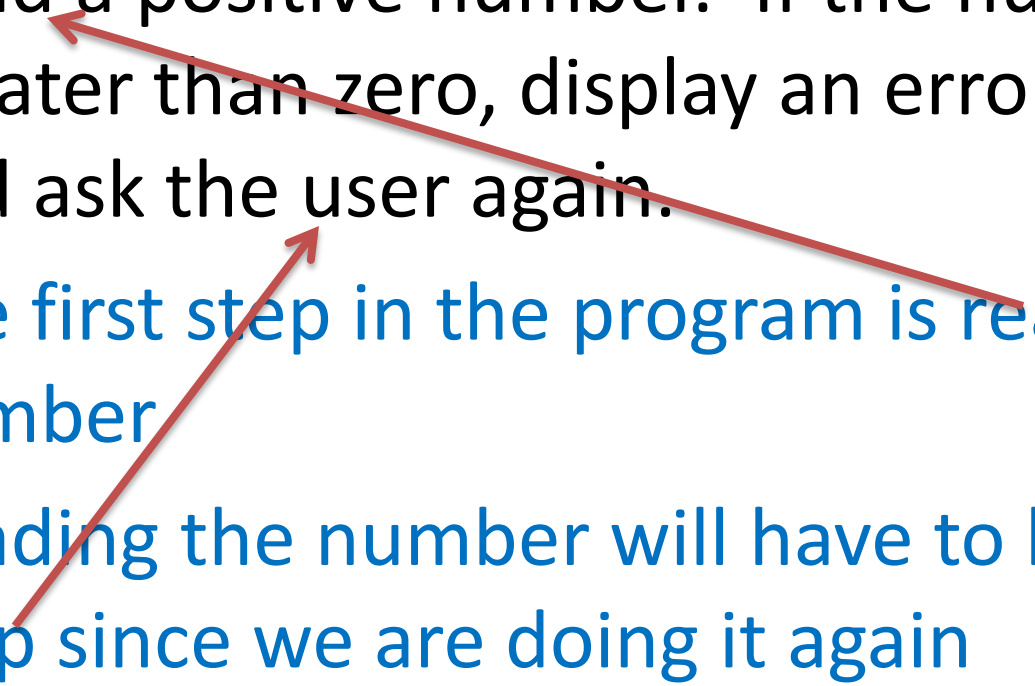
Analyzing the Problem

- Read a positive number. If the number is not greater than zero, display an error message and ask the user again.
- The program requires a decision, so there will be an Java if statement
- The program will do something multiple times, so there will be a loop



Analyzing the Problem

- Read a positive number. If the number is not greater than zero, display an error message and ask the user again.
- The first step in the program is reading a number
- Reading the number will have to be inside a loop since we are doing it again



Java Implementation

```
double number;
{
    System.out.print("Enter a number >");
    number = keyboard.nextDouble();
}
```

Analyzing the Problem

- Read a positive number. If the number is not greater than zero, display an error message and ask the user again.
- The java **`if`** statement will test if the number is less than or equal to zero
- When the Java **`if`** is true *(that is number <= 0)* the program will display an error message
- Display is done with System.out.println

Java Implementation

```
double number;
{
    System.out.print("Enter a number >");
    number = keyboard.nextDouble();
    if (number <= 0) {
        System.out.println("Be positive!");
    }
}
```

Analyzing the Problem

- Read a positive number. If the number is not greater than zero, display an error message and ask the user again.
- The loop condition is the same as the **`if`** condition, $number <= 0$
- We always want to execute the loop at least once to read the number. Therefore a do while loop is appropriate

Java Implementation

```
double number;
do {
    System.out.print("Enter a number >");
    number = keyboard.nextDouble();
    if (number <= 0) {
        System.out.println("Be positive!");
    }
} while (number <= 0);
```

Problem Description

- For a 400 pixel wide by 600 pixel tall picture, set the Red intensity to zero using setRed(int x, int y, int intensity)

Analyzing the Problem

- For a 400 pixel wide by 600 pixel tall picture, set the Red intensity to zero using setRed(int x, int y, int intensity)
- There will have to be a loop for the 400 x pixels and a loop for the 600 y pixels

What Sets All Red Pixels to Zero?

click on the next slide

```
for( x = 0; x < 400; x++) {
  for( y = 0; y < 400; y++) {
    setRed(x,y,0);
  }
}
```

A

```
for( x = 0; x < 400; x++) {
  setRed(x,y,0);
}
for( y = 0; y < 400; y++) {
  setRed(x,y,0);
}
```

B

```
for( x = 0; x < 400; x++) {
  setRed(x,y,0);
  for( y = 0; y < 400; y++) {
    setRed(x,y,0);
  }
}
```

C

```
for( x = 0; setRed(x,y,0); x+400) {
  for( x = 0; setRed(x,y,0); x+600){
  }
}
```

D

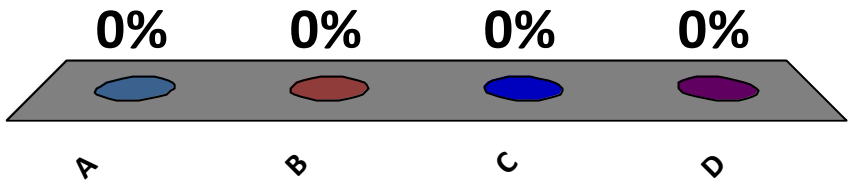
What Sets All Red Pixels to Zero?

A. A

B. B

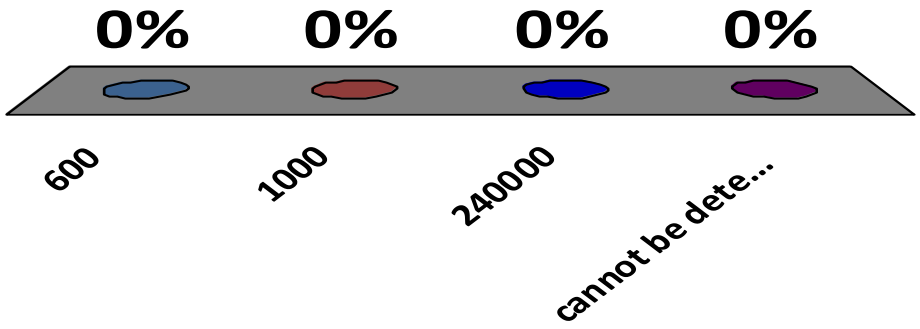
C. C

D. D



How many calls will be made to setRed?

A. 600
B. 1000
C. 240,000
D. cannot be determined



Problem Description

- Read the first ten integers from the file "numbers.txt" and display them on the screen

Analyzing the Problem

- Read the first ten integers from the file "numbers.txt" and display them on the screen
- The first steps are to create a Scanner object to read from the file
- The last step should be to close the file

Java Implementation

```
java.io.File dog = new java.io.File("numbers.txt");
Scanner cat = new Scanner( dog );




cat.close();
```

Analyzing the Problem

- Read the first ten integers from the file "numbers.txt" and display them on the screen
- There must be a loop that repeats 10 times
- Since we know how many times to loop, a `for` loops would be appropriate

Java Implementation

```
java.io.File dog = new java.io.File("numbers.txt");
Scanner cat = new Scanner( dog );

for (int counter = 0; counter < 10; counter++) {



}
cat.close();
```

Analyzing the Problem

- Read the first ten integers from the file "numbers.txt" and display them on the screen
- nextInt() can be used to read a number
- System.out.println can be used to display the number
- The read must come before the display

Java Implementation

```
java.io.File dog = new java.io.File("numbers.txt");
Scanner cat = new Scanner( dog );
int bull;
for (int counter = 0; counter < 10; counter++) {
        bull = cat.nextInt();
        System.out.println(bull);
}
cat.close();
```

Modify the Program

- Modify the program so that it will work correctly if there are less than 10 numbers in the file
- Work with your team to define the answer

Possible solution

```
java.io.File dog = new java.io.File("numbers.txt");
Scanner cat = new Scanner( dog );
int bull;
for (int counter = 0; counter < 10 && cat.hasNext();
                                        counter++) {
        bull = cat.nextInt();
        System.out.println(bull);
}
cat.close();
```

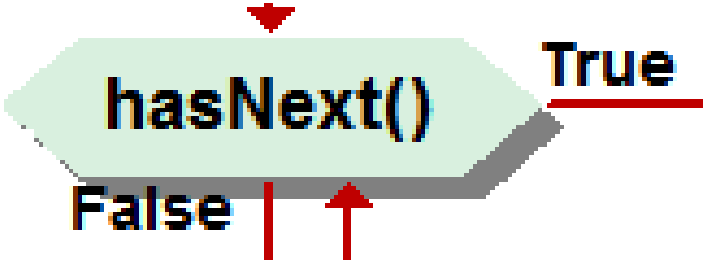
Another Possible solution

```
java.io.File dog = new java.io.File("numbers.txt");
Scanner cat = new Scanner( dog );
int bull;
loop: for (int counter = 0; counter < 10; counter++) {
        if (!cat.hasnext()) {
                break loop;
        }
        bull = cat.nextInt();
        System.out.println(bull);
}
cat.close();
```

Flowcharts

- Flowcharts are a way to visually describe the logic of a program
- It can be helpful to write a flowchart for a program and then convert the flowchart to Java

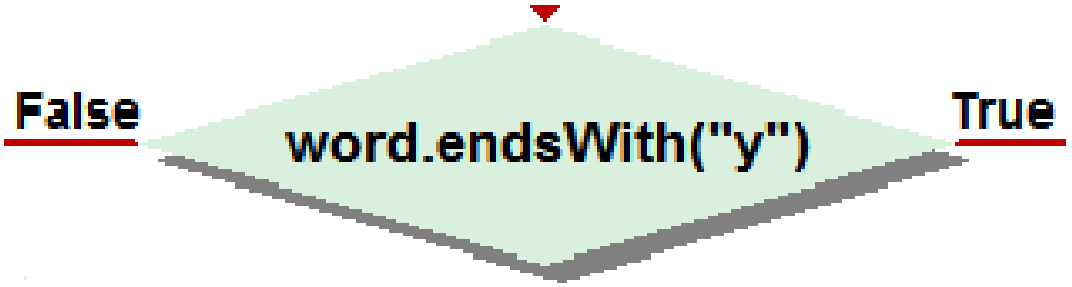
Flowchart to Java

- **input:word** converts to
  `word = keyboard.next();`
- **output: word** converts to
  `System.out.println( word );`
- 

converts to

```
while ( inFile.hasNext() )
```

Flowchart to Java



converts to

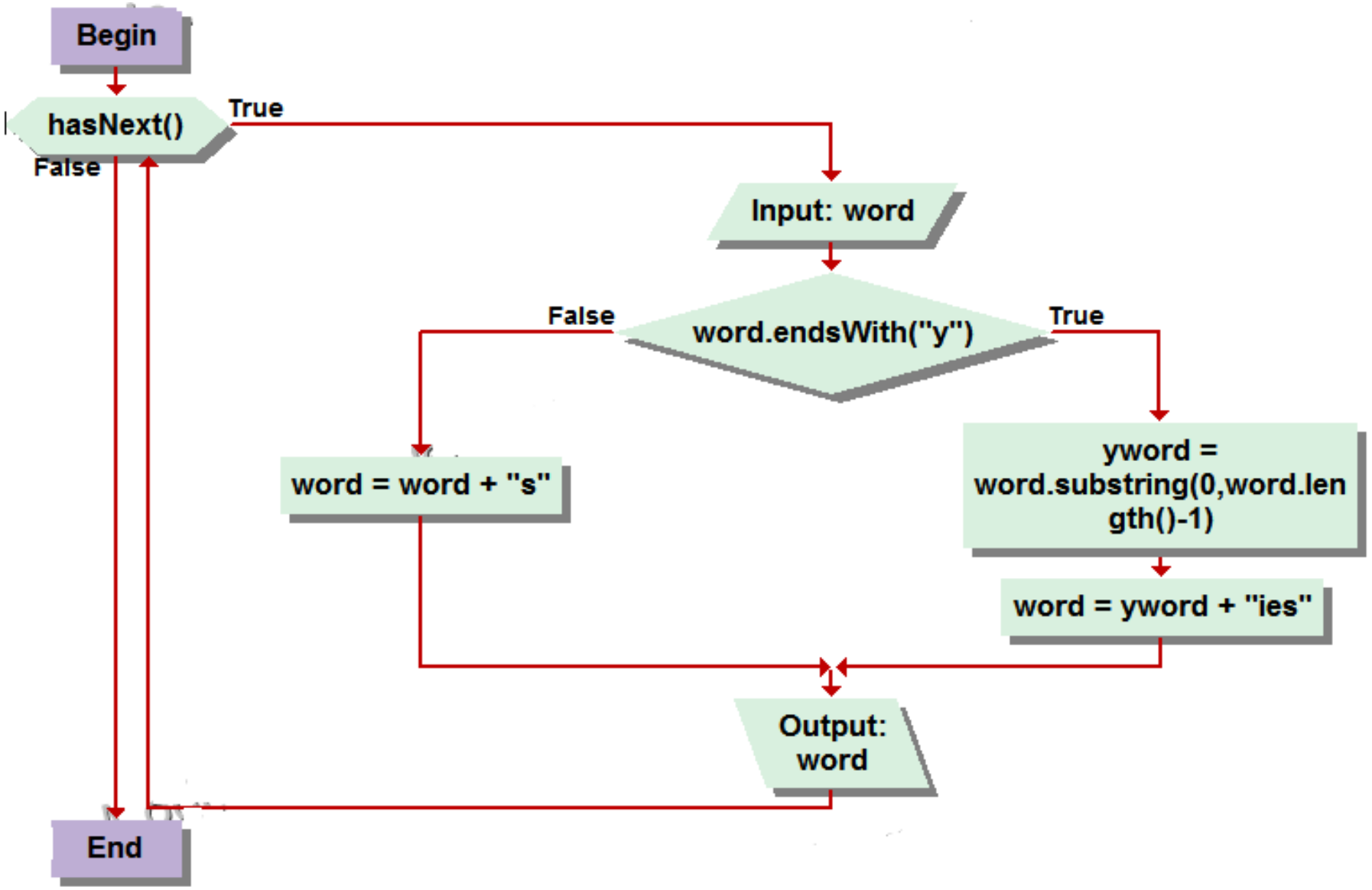
```
if (word.endsWith("y")) {
} else {
}
```

Problem Description

- Read an English word from a file and display the plural of the word
- For most words, you just add an "s" on the end
- If the word ends in "y", you change the "y" to "i" and add "es"

- *We will ignore the **many** other rules*

Making a word plural



Complete the Program

- Working with your team, complete this program to display the plural of the words

```
java.io.File dog = new java.io.File("data.txt");
Scanner cat = new Scanner( dog );
String word, yword;
```

Possible Solution

```
while (cat.hasNext()) {
    word = cat.next();
    if ( cat.endsWith( "y" )) {
        yword = cat.substring( 0, cat.length()-1 );
        word = yword + "ies";
    } else {
         word = word + "s";
    }
    System.out.println( word );
}
```

format Method

- The java.io.PrintWriter class has two identical methods, format and printf, to format output

  format(String format, var1, var2, …)

- Writes the variables to the output as specified by the format string
- Very similar to printf in the C programming language

Format Descriptors

- The format string may contain text with descriptors located in it.
- The descriptors start with a percent sign, % followed optionally by a length and then a format type character

format	data type	result
'd'	int or long	The result is formatted as an integer
'f'	double or float	The result is formatted as a decimal number
‘s’	String	The string

Output length

- You can specify a number between the % and the descriptor character to indicate the minimum number of characters to print
- You can specify the maximum number of digits to the right of the decimal point

%*minlength*.*maxprecision***f**

Format Examples

```
double   e = 2.718281828459045;
System.out.format("answer is %5.3f", e);
```

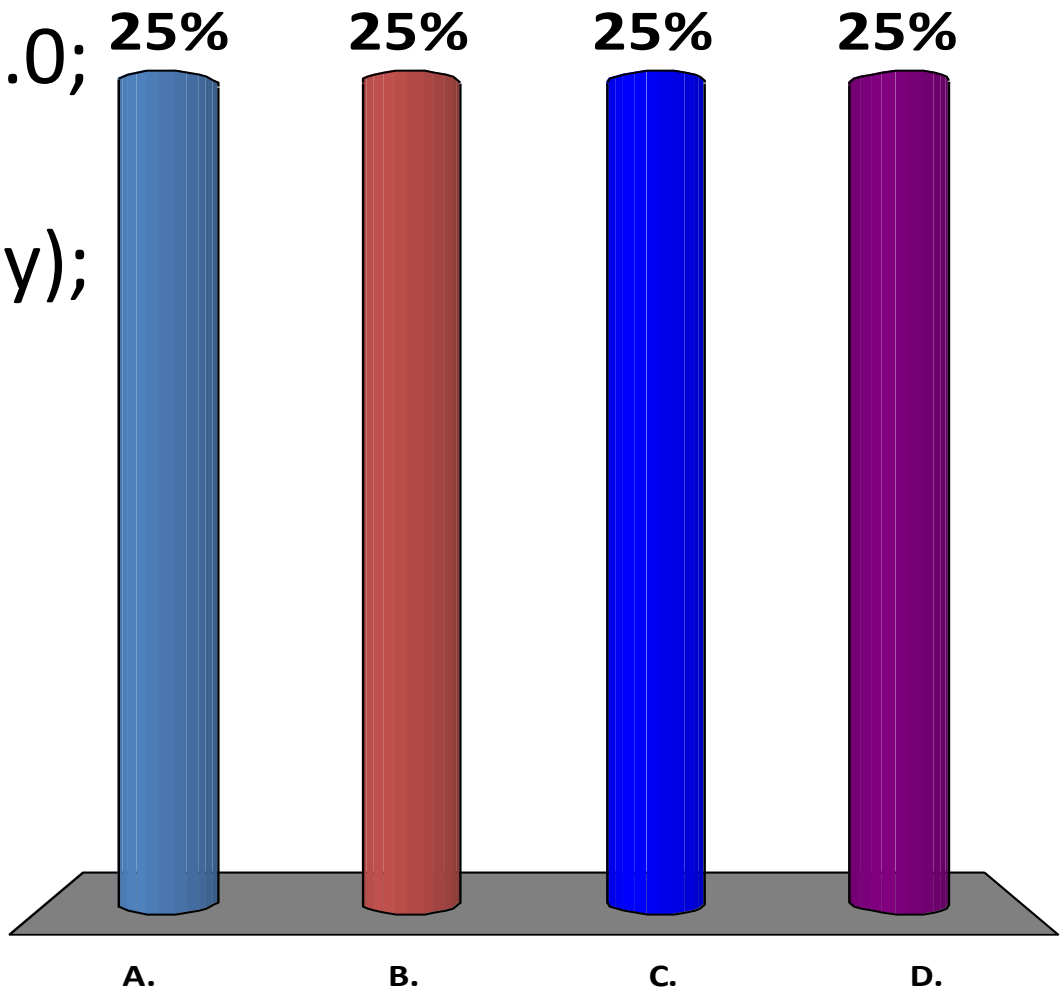
will display “answer is 2.718”

```
System.out.format("answer is %7.4f", e);
```

will display “answer is 2.7183”

What is displayed?

```
double x = 10.0;
double y = x * 2.0 / 3.0;
System.out.printf(
        "y is %6.3f", y);
```



A. y is 10.000

B. y is 6.666

C. y is 6.667

D. y is 6.66666666

Write the printf statement

```
double bill, tax;
tax = cost * 0.07;
bill = cost + tax + handling;

// Display the bill with two decimal digits
```

Founders' Day

- The Founders' Day Convocation is Thursday, March 21 from 10:00 – 12:00 in the Harrison Auditorium
- Classes are suspended during that time

- The lab quiz will be next week instead of this week

Exam

- The second exam will be in lecture on Friday, March 22
- The exam will cover everything since the first exam
    - If statements
    - Loops
    - Files
- A sample exam is available on Blackboard under course materials