COMP 110-001
Flow of Control: Branching 2

Yi Hong

May 19, 2015

Review

- if … else …
  - Q1: Write a program that
    - Reads an integer from user
    - Prints "Even" if the integer is even
    - Otherwise, prints "Odd"
- Boolean expression & Comparison
  - Q2: How to compare values of primitive types? How about objects?
  - Q3: Write the boolean expression for testing a leap year
    - A leap year is divisible by 4, but not by 100 (except if divisible by 400)

Today

- More if / else statements
- The switch statements

Gotcha (Syntax)

- if (boolean expression);

DO NOT DO THIS!!!!!!!

- if (boolean expression) {

  ….

  } else { // NO boolean expression here

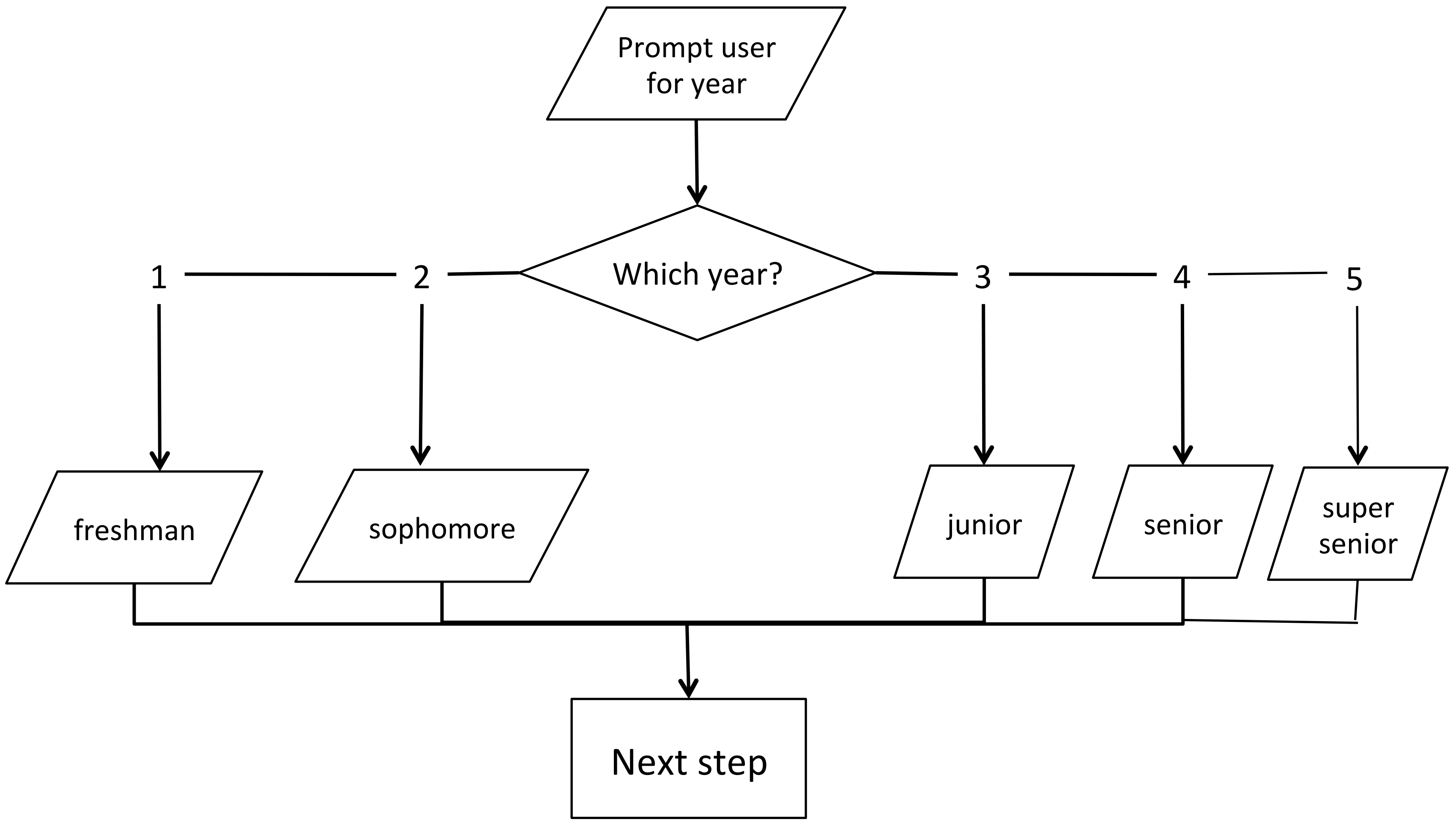
  ….

  }

Tracing if / else code

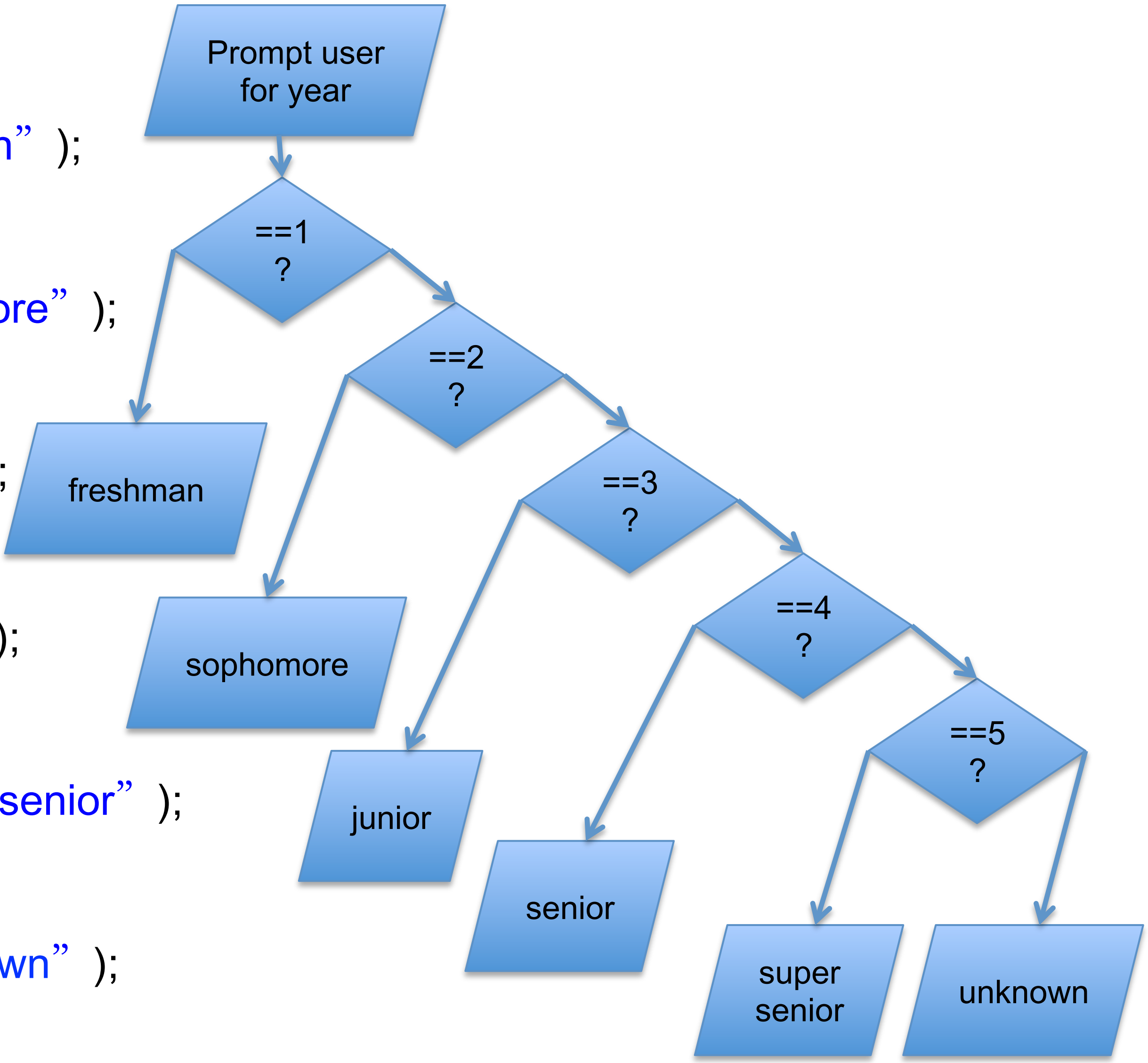
- Simple if-else statement: 2-choose-1
- Nested if-else statement: N-choose-1
  - Translate to multiple 2-choose-1 operations
- Example:
  - Write a program that takes an input, your year in college (as an integer), and outputs your year as freshman, sophomore, junior, senior, or super senior

Flow Chart

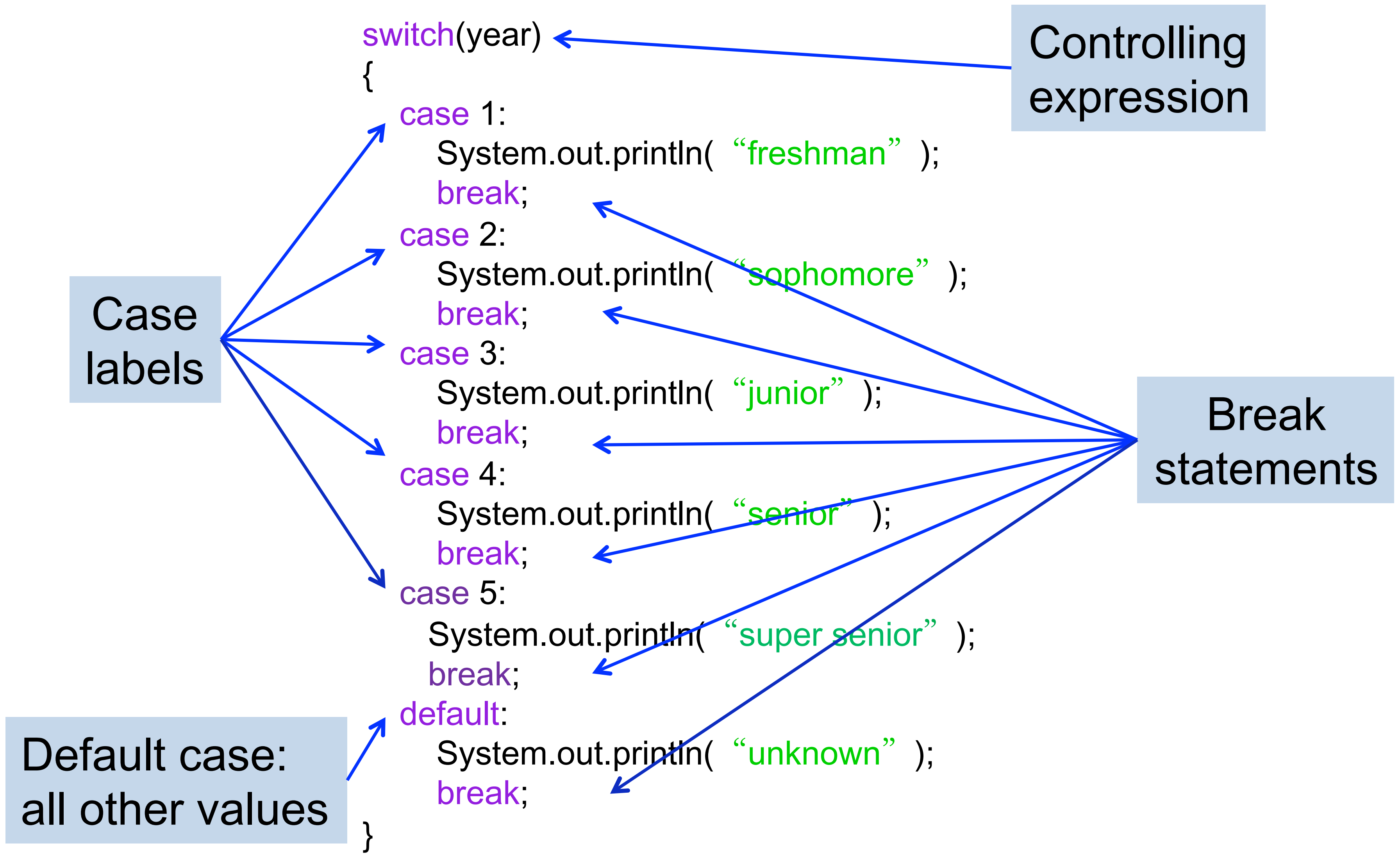

Prompt user for year
Which year?
1
2
3
4
5
freshman
sophomore
junior
senior
super senior
Next step

With Nested if / else

```
if (year == 1)
    System.out.println( “freshman” );
else if (year == 2)
    System.out.println( “sophomore” );
else if (year == 3)
    System.out.println( “junior” );
else if (year == 4)
    System.out.println( “senior” );
else if (year == 5)
        System.out.println( “super senior” );
else
        System.out.println( “unknown” );
```



Switch Statement



```
switch(year)
{
   case 1:
      System.out.println( “freshman” );
      break;
   case 2:
      System.out.println( “sophomore” );
      break;
   case 3:
      System.out.println( “junior” );
      break;
   case 4:
      System.out.println( “senior” );
      break;
   case 5:
     System.out.println( “super senior” );
     break;
   default:
      System.out.println( “unknown” );
      break;
}
```

Switch Statement Syntax

String has been supported in JDK 7

```
switch (Controlling_Expression)
{
   case Case_label:
      statements;
      break;
   case Case_label:
      statements;
      break;
   default:
      statements;
      break;
}
```

- Only int, char, enum can be used in the controlling expression
- Case labels must be of same type as controlling expression
- The break statement ends the switch statement, go to the next step outside the braces in the code
- The default case is optional

Practice with Switch Statements

- Write a switch statement that takes as the controlling expression the number of siblings a person has (as an int) and outputs an appropriate messages as follows:

Number of Siblings	Message
0	An only child
1	Just one you say
2	Two siblings!
3	Big Family!
4 or more	I don't believe you

Switch Statements

```
switch (numOfSiblings)
{
     case 0:
            System.out.print( “An only child” );
            break;
     case 1:
            System.out.print( “Just one you say” );
            break;
     case 2:
            System.out.print( “Two siblings!” );
            break;
     case 3:
            System.out.print( “Big family!” );
            break;
     default:
             System.out.print( “I don’t believe you” );
            break;
}
```

Enumerations

- Lists the values that a variable can have
  - E.g.: rate movies as either excellent, average, or bad
    - enum MovieRating {EXCELLENT, AVERAGE, BAD}
    - MovieRating rating;
    - rating = MovieRating.AVERAGE;
  - Provides a way to restrict the values of a variable
  - Actually a class, typically define it within another class, but always outside of method definitions. more discussion in Chapter 6

An Example

```
switch (rating)
{
     case EXCELLENT:
          System.out.print(  “You must see this movie!”  );
          break;
    case AVERAGE:
          System.out.print(  “This movie is OK, but not great.”  );
          break;
    case BAD:
          System.out.print(  “Skip it!”  );
          break;
    default:
          System.out.print(  “Something is wrong.”  );
          break;
}
```

- Avoid Logical Errors in the Multibranch if-else Statement

Order of Boolean Expressions

- What's the problem with the code below?

  -
    ```
    int n;
    if (n % 2 == 0) {
        System.out.println("Multiple of 2");
    } else if (n % 3 == 0) {
        System.out.println("Multiple of 3");
    } else if (n % 4 == 0) {
        System.out.println("Multiple of 4");
    }
    ```
- Ordering is important when boolean expressions are not mutually exclusive

Parallel & Mutually Exclusive Choices

- If the choices are mutually exclusive, we can write them as a list of if-only statements
- What’s the problem of doing this?

```
if (year==1) {
        System.out.println(“Freshman”);
}
if (year==2) {
        System.out.println(“Sophomore”);
}
if (year==3) {
        System.out.println(“Junior”);
}
```

An Example

- Determine the number of days in each month of a year

```
if (month == 4 || month == 6 || month == 9 || month == 11) {
        maxDay = 30;
} else if (month == 2) {
        boolean isLeapYear = (year % 4 == 0 && year % 100 != 0) || (year % 400
== 0);
        if (isLeapYear)  {
                maxDay = 29;
        } else  {
                maxDay = 28;
        }
} else {
        maxDay = 31;
}
```

The Conditional Operator

- Short-cut for if-else statements with return value

  boolean_expression ? value1 : value2 ;

  - If true, return value1; otherwise, return value2

  Example:

  ```
  if (n1 > n2)
        max = n1;
  else
        max = n2;
  ```

  can be written as

  ```
  max = (n1 > n2) ? n1 : n2;
  ```

- The ? and : together are call the *conditional operator* or *ternary operator.*

Some Tips

- Indentation in code makes it easy to read
  - Corresponds to “level” in code logic
- Eclipse can automatically fix indentation
  - Try Source → Correct Indentation or Format

Applications and Applets

- We will cover two kinds of Java programs: applications and applets
- Applications are regular programs, run on your computer
- Applets have graphical features, run within a Web browser

Next Class

- Lab 2 & 3
- About String, Java applets (Chapter 1.4), flow chart and if-else
- Bring your laptop and textbook