[TAP:JPBQN] Logic Operators

```
Student one;
if(one!=null && one.getYear() == 2){
   System.out.println(one.getName());
}
```

- Which of the following results from the code?
  - A. Compiler error
  - B. Run-time error
  - C. There's no error, but nothing is printed.
  - D. A name is printed.
  - E. Whatever

CSCI 136
Data Structures & Advanced Programming

Spring 2018

Instructors

Bill Jannen & Jon Park

Administrative Details

- Lab 1 handout is now online
- Prelab (=should be completed before lab):
  - Set up accounts
  - Complete Lab 1 design doc
    - Take a look at the example

Crash Course in Java

- Variables "content word" school nice ... x
  - Operators "function word" is +
  - Expressions "phrase" very nice x+3
  - Statements "sentence" x = x + 3;

Variable Types

- Primitive Types:
  - boolean, char, byte, short, int, long, float, double
- Objects :
  - arrays
    - Holds values of a single type
  - (class-based) Objects
    - Can hold information (fields)
    - Can specify behaviors (methods)

(General) Operators

- Unary
  - Arithmetic: +, -, ++, -- (prefix and postfix)
  - Logical: !
- Binary
  - Arithmetic: +, -, *, /, %
  - Relational: ==, !=, <, <=, >, >=
  - Logical: &&, ||
  - Assignment: =, +=, -=, *=, /=, %=
- Ternary
  - booleanCondition ? value1 : value2

Operator Precedence in Java

Operators	Precedence
postfix	*expr++ expr--*
unary	*++expr --expr +expr -expr* ~ !
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	|
logical AND	&&
logical OR	||
ternary	? :
assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

Crash Course in Java

- Variables
- Operators
- Expressions
- Statements

Expressions

2, "Jon" Math.pow(2,3)

Sequence of variables, literals, method calls & operators that evaluate to a value

- An expression returns a value
  - 3+2*5
  - "Your score is " + 100
  - i <= students.length boolean expression
  - (x++) x=2
  - (++x) x=2
- Note, an assignment expression also returns a value
  - y = 4 * (x = 3) // x=3 y=12
  - while ((line = reader.readLine()) != null){
    //do something with the line
    }

[TAP] Pre- and Post-increment

```
X = 10;
System.out.println(X++ * ++X);
X = 10;
System.out.println(++X * X++);
```

- Which of the following are outputted?
  - A. 100 & 100
  - B. 110 & 110
  - C. 120 & 121
  - D. None of the above
  - E. Whatever

Crash Course in Java

- Variables
- Operators
- Expressions
- ⊙ Statements

Statements

- Statements
  - `int x;`
  - `x = 3;`
  - `System.out.println("Hello, CS136!");`
  - `if (x > 3) { … } else { … }`
  - `while (x < 2) { … }`
  - `for (int i = 0; i < x; i++) { … }`

Control flow statement

Control Flow Statements

Select next statement based on a boolean expression.

- Branching structures: if, if/else, switch
- Looping structures: while, do-while, for

If/else

Example: Encode clubs, diamonds, hearts, spades as 0, 1, 2, 3

```
if (x == 0 || x == 2){
      System.out.println("Your card is red");
}else if (x == 1 || x == 3){
      System.out.println("Your card is black");}
else{
      System.out.println("Illegal suit code!");
}
```

switch

Example: Encode clubs, diamonds, hearts, spades as 0, 1, 2, 3

```
switch (x) {
   case 0:
   case 2:
      System.out.println("Your card is red");
      break;
   case 1:
   case 3:
      System.out.println("Your card is black");
      break;
   default:
      System.out.println("Illegal suit code!");
      break;
}
```

while & do-while

Example: Count # of flips until “heads”

```
Random rng = new Random();
int flip, count = 0;
flip = rng.nextInt(2); // returns 0 or 1
count++;
while (flip == 0) {
      flip = rng.nextInt(2);
      count++;
}
```

VS

```
do {
      flip = rng.nextInt(2);
      count++;
} while (flip == 0);
```

for & for-each

Example: Compute the average of test scores

```
int[] grades = { 100, 78, 92, 87, 89, 90 };
int sum = 0;
int i = 0;
while ( i < grades.length ) {
    sum += grades[i];
    i++;
}
ave = sum/grades.length;
```

while (exp)

⇓

for (; exp ;)

VS

```
for( int i = 0; i < grades.length; i++ )
    sum += grades[i];
```

VS

```
for (int g : grades)
    sum += g;
```

break & continue

break (Print the first prime between 100 and 200)

```
for( int i = 100; i <= 200 ; i++ ){
    if ( isPrime(i) ) {
        System.out.println( i );
        break;
    }
}
```

continue (Print all even # between 100 and 200)

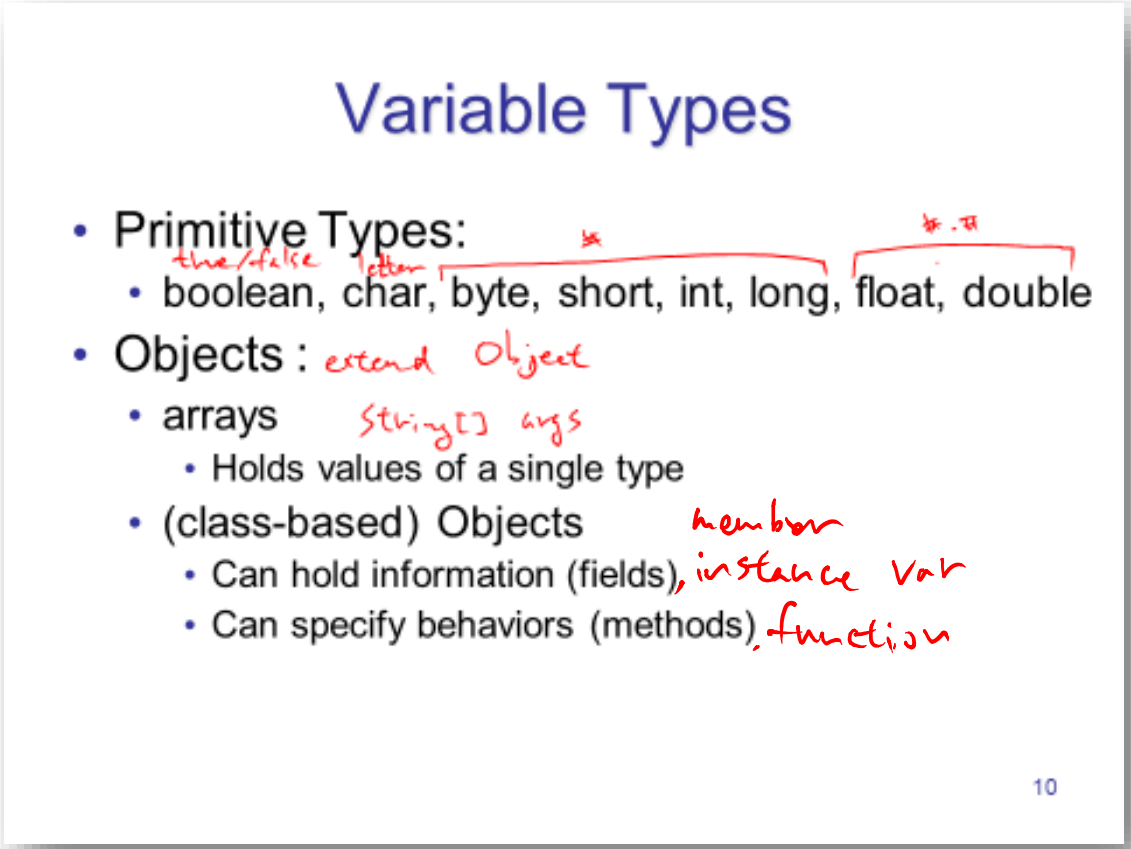
```
for( int i = 100; i <= 200 ; i++ ) {
    if (i%2 = 1){
        continue;
    }
    System.out.println( i );
}
```

Crash Course in Java

- Variables
- Operators
- Expressions
- Statements
- Object-oriented Programming

Object-Oriented Programming

- OOP is a programming paradigm, where a program is a set of objects interacting with one another.



OOP Example

Goal: Keep track of babies at a nursery; for each baby, keep this info: name and age.

Non-oop

String[] names

int[] age

oop

Baby

name

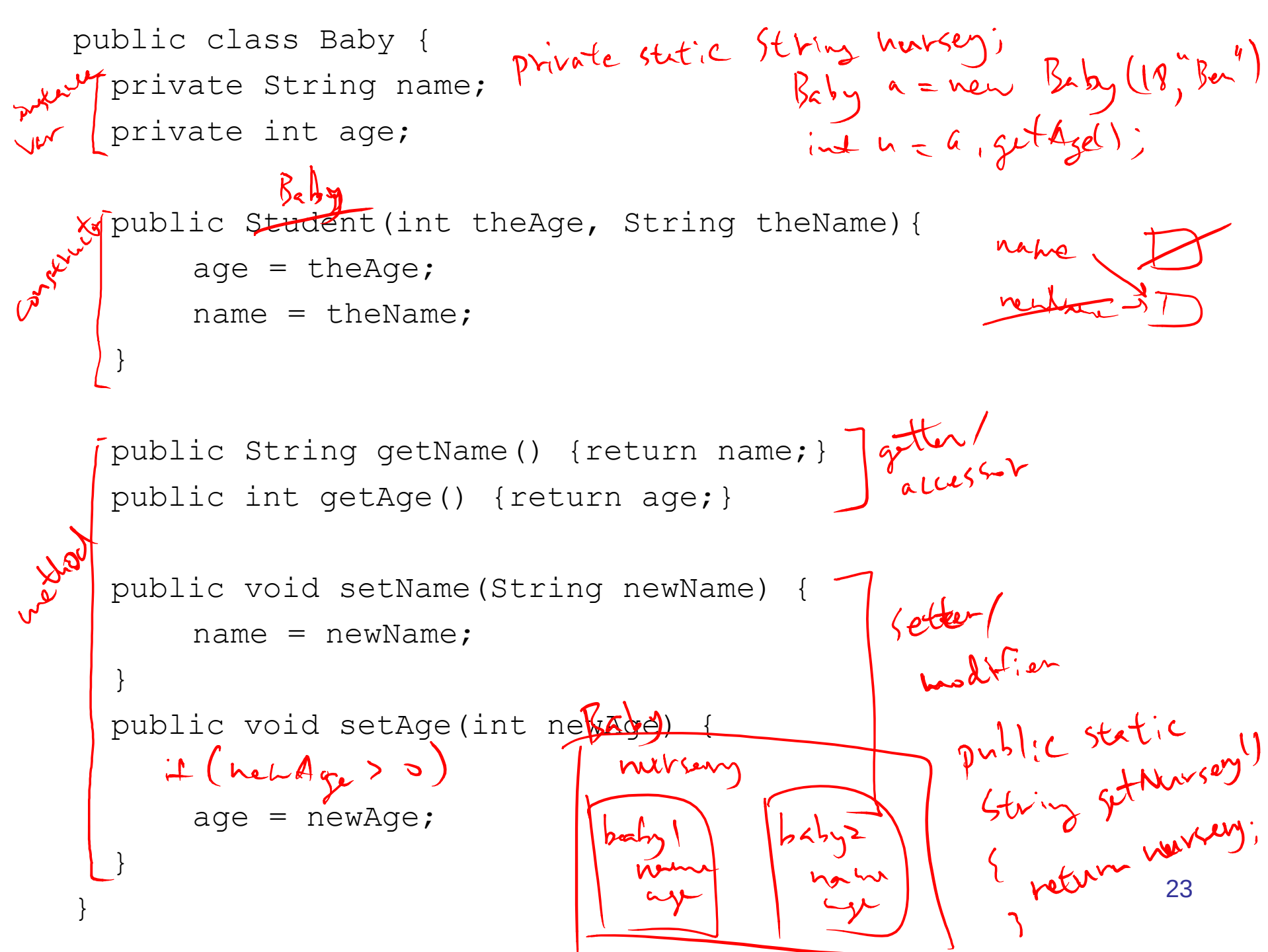
age

```
public class Baby {
    private String name;
    private int age;

    public Student(int theAge, String theName){
        age = theAge;
        name = theName;
    }

    public String getName() {return name;}
    public int getAge() {return age;}

    public void setName(String newName) {
        name = newName;
    }
    public void setAge(int newAge) {
        if (newAge > 0)
            age = newAge;
    }
}
```



Access Modifiers

	Same Class	Class in the Same Package	Any Subclass	Any Class
public	Y	Y	Y	Y
protected	Y	Y	Y	N
None (package)	Y	Y	N	N
private	Y	N	N	N

Again, be as restrictive as possible!

Revisiting Hello.java

```
public class Hello {
  public static void main(String[] args){
       System.out.println("Hello, CS136!");
  }
}
```